2

Evaluating Intelligence Systems That Support Deep Fires

Frank Camm, Norman Z. Shapiro,
Robert H. Anderson, James J. Gillogly,
Jean LaCasse, Mark LaCasse

RAND | ARROYO CENTER

The research described in this report was sponsored by the United States Army, Contract No. MDA903-86-C-0059.

Library of Congress Cataloging in Publication Data

Evaluating intelligence systems that support deep fires / Frank Camm
... [et al.].
p. cm.
"Prepared for the United States Army."
"October 1989."
Includes bibliographical references.
"R-3795-A."
ISBN 0-8330-1043-3
1. Military intelligence—United States—Computer simulation.
2. Air warfare—Computer simulation. I. Camm, Frank A., 1949-
II. RAND Corporation. III. United States. Army.
IV. Title: Deep fires.
UB251.U5E95 1990
355.3'432'0973—dc20 90-33231
CIP

The RAND Publication Series: The Report is the principal publication documenting and transmitting RAND's major research findings and final research results. The RAND Note reports other outputs of sponsored research for general distribution. Publications of The RAND Corporation do not necessarily reflect the opinions or policies of the sponsors of RAND research.

Published by The RAND Corporation
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

R-3795-A

Evaluating Intelligence Systems That Support Deep Fires

Frank Camm, Norman Z. Shapiro,
Robert H. Anderson, James J. Gillogly,
Jean LaCasse, Mark LaCasse

October 1989

Prepared for the
United States Army

RAND

PREFACE

This report documents the results of the "C^3I Architectures" project, performed within the Force Development and Employment Program of the Arroyo Center. This program studies the coordinated use of new weapon, intelligence, and command and control systems for effective deep fires against the Soviet second echelon in Central Europe. It examines U.S. combat intelligence systems in this broader context, asking how the contribution of an intelligence system to the effectiveness of deep fires can be measured.

The report explains a methodology RAND has developed to evaluate intelligence systems supporting deep fires. The methodology is implemented in C language and RAND-ABEL code suitable for use on a Sun 4 workstation. The code generates high-quality graphics that a user can exploit to examine results and interact with the evaluation system. The code has not been used for any real evaluations to date. Keywords:

The principal audience for this report are the analysts who are responsible for developing effective concepts and doctrine on deep fires. It should also interest analysts and decisionmakers responsible for modeling and evaluating the effectiveness of intelligence systems. Although the focus is on a combat intelligence system with a specific mission in Central Europe, the report will interest analysts attempting to simulate intelligence fusion without becoming lost in the mass of detail that intelligence systems must process and those more generally concerned with rule-based simulations and simulations based on Bayesian logic. The study offers several basic innovations in fusion modeling.

Command control communications; Military intelligence (RH)

THE ARROYO CENTER

The Arroyo Center is the U.S. Army's federally funded research and development center for studies and analysis operated by The RAND Corporation. The Arroyo Center provides the Army with objective, independent analytic research on major policy and management concerns, emphasizing mid- to long-term problems. Its research is carried out in five programs: Policy and Strategy; Force Development and Employment; Readiness and Sustainability; Manpower, Training, and Performance; and Applied Technology.

Army Regulation 5-21 contains basic policy for the conduct of the Arroyo Center. The Army provides continuing guidance and oversight through the Arroyo Center Policy Committee, which is cochaired by the Vice Chief of Staff and by the Assistant Secretary for Research, Development, and Acquisition. Arroyo Center work is performed under contract MDA903–86-C-0059.

The Arroyo Center is housed in RAND's Army Research Division. The RAND Corporation is a private, nonprofit institution that conducts analytic research on a wide range of public policy matters affecting the nation's security and welfare.

Stephen M. Drezner is vice president for the Army Research Division and director of the Arroyo Center. Those interested in further information concerning the Arroyo Center should contact his office directly:

Stephen M. Drezner
The RAND Corporation
1700 Main Street
P. O. Box 2138
Santa Monica, California 90406-2138

Telephone: (213) 393-0411

SUMMARY

Current U.S. Army doctrine emphasizes the importance of extending command emphasis to include not just the close battle but the deep battle. It calls for the use of deep fires and maneuver to exploit the deep portion of the battlefield. Planning focuses on generating deep fires with Air Force aircraft and the Army Tactical Missile System (ATACMS). The quality of intelligence on the deep battlefield can greatly affect the performance of both types of deep-fire assets. The U.S. Army and Air Force are both pursuing alternatives that would enhance their intelligence on the deep battlefield during combat. Neither has an integrated way to ask how specific changes in the U.S. intelligence system would affect their ability to execute deep fires.

This report presents an analytic approach that could simulate the development of combat intelligence about the deep battlefield and compare the performance of alternative intelligence systems to support deep fires. It emphasizes the development of intelligence products that the Army could use to support the ATACMS in a Central European war in the mid-1990s. It draws on observations of combat intelligence activities during several U.S. and NATO command-post exercises in Germany during 1986–1988 and on Army-approved European scenarios and Army combat and intelligence collection models to provide inputs to the simulation of the intelligence system as a whole. It also uses measures of performance of greater interest to a combat commander and his staff than to the intelligence community itself.

The analytic approach presented here employs a set of new techniques for modeling the quality of information and changes in the quality of information in an intelligence system. It uses simple Bayesian logic to develop a high-level view of intelligence processing and realizes it in a flexible, parameterized, rule-based network model. Although the model is tailored to the problem at hand, the techniques could be applied in a broader context to a wide range of questions about the performance of intelligence systems.

This approach views combat intelligence in the context of a "system of systems." An *intelligence* system comprises *collection*, *processing*, and *communication* systems. U.S. Army doctrine places the corps at the center of the intelligence system responsible for deep battle. This approach can model the collection, processing, and communication systems that are both organic to a corps intelligence system and that a corps depends on to develop and distribute combat intelligence products. The Army can vary the system by varying these component

parts. This report provides a flexible, transparent way to measure how changes in individual systems affect the performance of the intelligence system as a whole, using high-quality graphics to display information.

Briefly, the approach works in the following way:

1. Fix a baseline in two steps. (a) Over the course of an engagement, use military judgment to set assumptions about the behavior of Red[1] units on the deep battlefield, Blue priority information requirements (PIRs), the Blue collection schedule, and delays in processing and communication. (b) Given these assumptions, simulate how new information on Red behavior moves through the intelligence system, updating various databases in the system and ultimately influencing the quality of information available to Blue commanders and their staffs.
2. Incrementally change the presence, use, or performance of a constituent part of the intelligence system. Given the assumptions in step (a) of the baseline, rerun the simulation to determine the quality of information available to Blue commanders following the change.
3. Compare the quality of information on the Red order of battle available to commanders (a) in the baseline and (b) following a change, to determine the effect of the change on the performance of the intelligence system as a whole.

This approach to evaluation and simulation differs from other approaches in six important ways.

1. *We use the quality of intelligence products as a figure of merit and, among these products, focus on the Red order of battle in the deep battlefield.* Other approaches look at the technical performance of parts of an intelligence system, time lines for delivering information from the battlefield to a decisionmaker, and the effect of intelligence development on combat outcomes. All of these measures are valid and useful in particular applications. Our measure is best for looking in depth at the performance of the intelligence system as a whole without having to determine how it interacts with other sources of combat capability.

2. *We look at incremental changes in intelligence systems.* Our approach allows us to examine how certain changes in specific components of an intelligence system affect its total performance. Focusing on incremental changes allows us to avoid the ambiguities involved in modeling important feedbacks that military intelligence decisionmakers and analysts do not understand very well. These include

[1] Throughout the report, we use "Red" to refer to the enemy and "Blue" to refer to friendly forces.

feedbacks within an intelligence system and between the intelligence system and other combat capabilities. For example, we need not posit assumptions about how information flows affect delays in communication and processing, how information available today affects the demand for information tomorrow, or how changes in the quality of information affect combat outcomes and hence the nature of Red behavior in the future. The last point also means that we need not posit assumptions about how an intelligence system transforms data on the Red order of battle into higher-level inferences about Red intentions or how these inferences affect command decisions. Therefore, where there are great uncertainties, we can use issues where more is understood to draw more easily defensible conclusions about the performance of combat intelligence systems. Where feedbacks like those that we avoid are important to policy, however, a user should consider an alternative approach that looks beyond the effects of incremental changes.

3. *We rely heavily on Army models for input.* As currently formulated, our approach relies on the Army's Vector-in-Commander (VIC) corps combat model to simulate the behavior of Red units on the deep battlefield and aspects of a Blue intelligence system's collection of data on this behavior. Given its status as the Army's approved corps combat model, VIC embodies Army doctrine in a way that no other available model does. Furthermore, to our knowledge, no other models provide VIC's depth of detail; we need that detail to provide the richness we seek in our own simulation. We have adjusted inputs from VIC in small ways that improve its output without challenging Army doctrine. With certain modifications, those who prefer a combat simulation other than VIC can use their own combat simulations to drive our simulation.

4. *We emphasize simulating the quality of intelligence products, not the generation of these products per se.* One intuitively appealing way to present information about the performance of an intelligence system might be to simulate the Red order of battle as Blue intelligence perceives it, compare this perception with the true Red order of battle, and use the difference between the two as a figure of merit. We rejected this potentially attractive approach because simulating a perceived Red order of battle would require massive detail on specific fusion rules and strong assumptions regarding higher-level inferences about Red intentions and behavior; such a simulation cannot disentangle the order of battle from these higher-level inferences. Past attempts to simulate a perceived Red order of battle have yielded enough questionable inferences to undermine confidence in the simulations. Our method simulates the quality of intelligence directly without attempting to simulate specific perceptions.

5. *Our simulation of intelligence fusion takes a high-level approach to avoid getting lost in the intricate detail of true fusion.* As a result, our approach does not attempt to collect rules that order-of-battle analysts and automated systems use to execute fusion and provide an inference engine that executes these rules together. Our approach is based on a set of intelligence concepts and parameters that we have not seen in earlier simulations of intelligence development. For example, whereas past efforts have typically used a probability of detection to measure the quality of information yielded by collection, our approach uses a likelihood ratio that measures the potential ability of a sighting of Red behavior to discriminate between competing hypotheses. This concept and others like it have analytic power and ability to capture basic ideas underlying the detailed rules of thumb used in true fusion. Because other analysts have not used these concepts in the past, no one has attempted to collect data to measure them, thus complicating the immediate implementation of our approach. If, as we expect, our approach simplifies the effective simulation of fusion at a high level, the data we need should become more accessible, making our approach easier to implement as time passes.

6. *We implement our approach with a computer code that promotes easy understanding and modifications to include new rules as needed.* Substantive portions of the code are in the C-based, English-like language RAND-ABEL, which allows users with little programming experience to look directly at the implemented code and understand what it is doing. The structure of the code makes it easy to change collectors, processors, communications links, and the way they interact in an intelligence system. Its clear, modular form allows targeted adjustments in the code if new rules are required to characterize specific capabilities important to a policy evaluation. The structure of the model uses newly developed methods for tracing changes in the quality of information while the model operates; they facilitate the development of simulations and the analysis of the information they generate. RAND-ABEL also gives us access to the editing and graphics utilities of the RAND Strategic Assessment System (RSAS), utilities that facilitate this approach and the interpretation of its output.

Although the principal purpose of the model presented here is to facilitate evaluation of intelligence systems that support deep fires, the analytic techniques developed to implement this model should have much broader application:

- The measure of information quality that we use, the subjective probability that a component of the intelligence system implicitly associates with the true value of an attribute of a Red unit, markedly simplifies the use of Bayesian methods to model information quality.

- The ways that we (a) measure the information content of new sightings of Red units on the battlefield and (b) model changes in the quality of information in databases in the absence of new sightings allow the simple application of a robust Bayesian model of information updating.
- These Bayesian techniques provide the key to simulating the quality of information without simulating the content of the information itself. This reduces the data required for modeling and increases the credibility of analytic results.
- The incremental approach to evaluation applied here substantially reduces the set of assumptions, hence the amount of sensitivity analysis required to model intelligence development.
- Our new modeling devices increase the transparency and flexibility of the formal model and computer code that implement our simulation.

Each of these innovations could find broader application in the simulation of intelligence development and in the development of rule-based simulation in general.

This approach provides a flexible, accessible analytic environment in which to simulate the quality of information produced by a corps intelligence system and intelligence assets associated with it. Using this environment for policy analysis requires collecting data with which to calibrate simulations for specific applications. As experience with the approach accumulates, calibration should become simpler and the range of potential applications for the approach should grow. Also, many of the new analytic techniques that underlie our approach should find application in the analysis of other kinds of problems.

ACKNOWLEDGMENTS

RAND colleagues Bernard Rostker and Bruce Goeller conceived this project. Lewis Jamison participated actively in gathering information on intelligence activities early in the project. Kenneth Solomon has helped administer portions of the project. Sinclair Coleman, Edward Hall, and James Kahan reviewed and provided detailed comments on an earlier draft. Many other RAND colleagues provided comments and discussions, especially Patrick Allen, John Bondanella, Edison Cesar, David Rubenson, Peter Stan, Cathleen Stasz, and Robert Worley. Many people in the Army also contributed to the success of the project. It began under the oversight of Colonel Daniel Whiteside. Thomas Douthitt acted as project monitor for most of its life. Colonel Kenneth Carlson acted as principal U.S. Army point of contact in Europe. Colonels Duane Myers and Harold Sprague helped gather information in CENTAG and 4ATAF. Colonel Rudolph Peksens (USAF) provided access to exercises at the Warrior Preparation Center. Army and Air Force personnel at all levels in V Corps, VII Corps, USAREUR, CENTAG, and 4ATAF in Europe and at Headquarters, Department of the Army, and numerous locations in the Army Training and Doctrine Command freely gave interviews. The authors retain responsibility for any errors of fact or interpretation.

CONTENTS

Section

FIGURES

TABLES

GLOSSARY

ASAS/ENSCE. All-Source Analysis System/Enemy Situation Correlation Element. New joint U.S. Army-Air Force system for rapidly integrating information from different intelligence systems, transforming it into useful products, and disseminating it to users.

ASPS. All-Source Production Section. The organization within the CTOC Support Element currently responsible for integrating information from different intelligence disciplines to generate a Red order of battle, situation assessment, and target list.

ATACMS. Army Tactical Missile System. A new weapon system that will enhance the Army's ability to generate deep fires when it is introduced in the 1990s.

Blue. Pertaining to friendly forces, activities, or capabilities.

C. A computer programming language in which portions of the PRO model are implemented.

CENTAG. Central Army Group. The NATO Army organization that coordinates and directs the defense of southern West Germany.

COMINT. Communications Intelligence. Intelligence based on the "external" signatures of radio transmissions or their "internal" content.

CTOC. Corps Tactical Operations Center. The staff immediately responsible for the coordination and direction of U.S. Army corps operations.

Decibel. One-tenth of a bel. A unit used to measure ratios on an additive scale.

Discrimination Ratio (DR). A likelihood ratio that expresses the information content of a sighting of a Red unit-attribute in terms of the information's ability to discriminate between two hypotheses that the sighting was generated by (a) the true value of this unit-attribute and (b) some other value.

EACIC. *Echelon-Above-Corps-Intelligence Center*. The organization within U.S. Army in Europe (USAREUR) responsible for intelligence development on the area of the battlefield beyond the corps areas of responsibility.

ELINT. Electronic Intelligence. Intelligence based on the signatures of radar transmissions.

Enhancement. Change since the beginning of simulation of $-10 \times \log_2$ OR, when OR is the odds ratio, $(1 - P)/P$, and P is the subjective probability that Blue associates with the true value of the unit-attribute to which this enhancement applies.

Enhancement Increment (EI). $-10 \times \log_2$ DR, when DR is the discrimination ratio associated with a unit-attribute.

FLOT. Forward Line of Own Troops. Loosely speaking, the forwardmost line of Blue positions, hence the dividing line between Red and Blue forces.

4ATAF. Fourth Allied Tactical Air Force. The NATO Air Force organization that coordinates and directs the defense of southern West Germany.

FSCL. Fire Support Control Line. The dividing line between the division and corps areas of responsibilities.

GRCS. Guardrail Common Sensor. A new U.S. Army standoff intelligence platform that will carry COMINT (Guardrail) and ELINT (Quicklook) sensors when it is introduced in the 1990s.

Ground Truth. A description of the time-dependent status of Red and Blue forces as determined by a simulation external to the PRO model.

HUMINT. Human Intelligence. Intelligence from human sources including, among other things, penetrating patrols of Blue forces, refugees, prisoners of war, and spies.

IMINT. Imagery Intelligence. Intelligence based on imagery including, among other things, imagery generated by photographic, video, radar, and infrared means.

JSTARS. Joint Surveillance, Target Acquisition, and Reconnaissance System. A joint U.S. Army-Air Force standoff platform that uses radar to collect and distribute information on moving and fixed objects in near-real time.

MTI. Moving Target Indicator. A sensor that uses radar to detect moving objects, especially trucks, tanks, and other heavy vehicles.

Message. In the context of PRO, a device that carries a quantum of information on a unit-attribute through an intelligence system until the information ultimately becomes embodied in final intelligence products.

Node. In the context of PRO, a collector, processor, or commander, each of which is represented as a node in the network model that implements PRO.

Observation. A PRO message regarding a particular unit-attribute containing an enhancement increment representing the quality or completeness of that unit attribute.

Order of Battle. The status of key attributes of Red units, including their identity, type, echelon, location, speed, direction of movement, effectiveness level, and activity.

Order of Battle Database. A PRO database containing time-varying attributes with their intelligence qualities of all modeled Red units perceived at a particular Blue node.

PIR. Priority Information Requirement. A regularly generated statement of the commander's highest-priority information needs.

Pre-Observation. A PRO message regarding a particular unit-attribute arising from a sighting provided by the ground truth simulator, containing a measure of the quality or completeness of that unit-attribute.

PRO. Intelligence Propagation Model. A computer model of the propagation and quality variation of intelligence.

RAND-ABEL. A computer programming language developed at RAND that is easy to read by persons unfamiliar with it. Most of the PRO model is programmed in RAND-ABEL.

Red. Pertaining to enemy forces, activities, or capabilities.

RIPL. Reconnaissance Interdiction Planning Line. The line that divides the corps and echelons-over-corps areas of responsibility.

RSAS. RAND Strategy Assessment System. An automated wargaming facility developed at RAND. Portions of the RSAS system were used or adapted for use in the PRO model.

SSM. Surface-to-Surface Missile.

UAV. Unmanned Aerial Vehicle. An automated airborne vehicle that can act as a penetrating platform for IMINT, COMINT, or ELINT sensors.

Unit. A unit modeled in VIC or some other combat simulation used to drive PRO.

Unit-attribute. Any one of the order-of-battle attributes associated with a unit and modeled in PRO.

UNIX. A popular computer operating system available on most contemporary computers. The PRO model described in this report runs as an application program on a Sun Microsystems computer operating under UNIX.

VIC. Vector-in-Commander. The currently authorized U.S. Army corps combat simulation, which the Army uses to generate official scenarios. VIC was modified by project personnel to operate under UNIX.

I. INTRODUCTION

Current U.S. Army doctrine emphasizes the importance of extending command emphasis to include not just the close battle but the deep battle as well. Roughly speaking, the deep battlefield is the portion of the battlefield lying beyond line of sight of Blue observers at the forward line of own troops (FLOT). It is a portion of the battlefield that Blue can exploit only with enhanced intelligence assets. Doctrine calls for deep fires, maneuver, and electronic warfare to exploit the deep battlefield. Planning emphasizes the use of Air Force aircraft and the Army Tactical Missile System (ATACMS) to generate deep fires. The quality of intelligence on the deep battlefield can greatly affect the performance of both types of deep-fire assets. The U.S. Army and Air Force are both pursuing alternatives that would enhance their intelligence on the deep battlefield during combat. Neither has an integrated way to ask how specific changes in the U.S. intelligence system would affect their ability to execute deep fires effectively.

This report presents an analytic approach that could be used to simulate the development of combat intelligence about the deep battlefield and to evaluate the effects on performance of incremental changes in intelligence systems proposed to support deep fires. The report emphasizes the development of intelligence products that the Army could use to support the ATACMS in a Central European war in the mid-1990s. It draws heavily on observations of combat intelligence activities during several U.S. and NATO command-post exercises in Germany during 1986–1988 and relies on Army-approved European scenarios and Army combat and intelligence collection models to provide inputs to the simulation of the intelligence system as a whole. While carefully incorporating information from the Army intelligence community, it maintains a broader perspective, using measures of performance of greater interest to a combat commander and his staff than to the intelligence community itself. More generally, the methodology proposed could be applied in a much broader context to consider a wide range of questions about intelligence on the deep battlefield.

Section II explains our concept of an intelligence system as a "system of systems." It describes the basic functions of an intelligence system and how these work in NATO's Central Army Group (CENTAG) in Central Europe. Alternative ways to evaluate such a system and why we chose the proposed approach are given. An outline is provided of the simulation approach we use to compare the actual Red order of battle with that perceived by Blue intelligence. This section also describes the U.S. Army Vector-in-Commander (VIC) corps combat model that figures prominently in our simulation.

Section III explains how we model the basic functions of an intelligence system as components of a network, how we view information in terms of data on specific attributes of Red units on the deep battlefield, and how we model the movement of this information through such a network. Aspects of collection, processing, and communication delay the movement of this information in the network. Section III shows how we model these sources of delay. Finally, a simple example of an intelligence system illustrates how our approach would represent the movement of information in this system.

Section IV indicates how we measure and simulate changes in the quality of intelligence produced in an intelligence system as information moves through it, how formal concepts of subjective probability can represent uncertainty associated with battlefield information and its accuracy. Alternative analytic views of intelligence fusion are reviewed and is how a form of database updating simulates fusion. Finally, the section describes the extraction and processing of intelligence information from VIC to initiate fusion in the network we use to represent an intelligence system.

Section V uses a numerical illustration to show how the concepts explained in Sections III and IV work together. It first shows how we accept a set of unit sightings by this system from VIC and transform output from VIC into a usable form. Based on a posited set of delay and priority factors, information from these sightings flows through the simple intelligence system until it affects intelligence products relevant to a corps commander. Information from these sightings affects the quality of information available to the corps commander and the quality changes over time. Information about the quality of intelligence available to the commander permits evaluation of changes.

Section VI presents more technical information on the computer programs. It explains our formal model architecture, with its data structures and process control flows. It describes how we actually extract data from VIC, transform them, and place them in appropriate data structures for our simulation. It explains the formal structure of PRO, the propagation model we use to transform information on the quality of inputs from sensors into information on the quality of Blue's Red order of battle. Analysts can manipulate and display data on the quality of this order of battle to support policy analysis.

Section VII concludes the report with some observations about the general problem of simulating intelligence and its effect on combat outcomes. It addresses the basic problem of using output from such a simulation to calibrate it. Other applications of the approach presented here are also discussed.

The appendix describes aspects of deception that can be included in our approach. We have not yet implemented them in detail.

The reader will detect three different perspectives in this report. The first concerns the general problem of simulating an intelligence system and the way we deal with that problem. The second concerns our choice of VIC as a source of information of the behavior of Red units on the deep battlefield and the collection of information about this behavior. Our propagation model can accept such information from alternative sources; but to use information from VIC, we must use certain specific forms. The third perspective concerns the simple intelligence system we use to motivate and illustrate arguments in the text. PRO, using inputs from VIC, can model much more complex intelligence systems with very different delay and priority factors from those in our simple example. Where it is unclear how broadly applicable a statement is in the text, we attempt to clarify which perspective applies to that statement.

II. BASIC ISSUES RELEVANT TO EVALUATING COMBAT INTELLIGENCE

A SYSTEM OF SYSTEMS

A combat intelligence system determines what raw information to collect, collects it, transforms it into usable intelligence products, and distributes these products to operators who can use them. Individual systems back up each of these tasks. A command and control system conveys information requirements to the intelligence community, which translates this into a specific collection plan. Collection systems—sensors, ground stations, surveillance units, and so on—execute the collection plan and convey raw information back to processors. Processing systems—correlators, database managers, order of battle shops, and so on—combine information from many sources and prepare internal products that provide the basis for intelligence reports. Production shops for situation assessment and targeting generate these reports. A distribution system sends these reports to operators who can use the information to support their combat plans and operations. A communication system supports all of these activities by moving information from one place to another. Specially arranged "quick fire channels" can move near-real-time information from collection systems to artillery units for immediate use. An "intelligence system" includes all of these systems and their interactions.

An intelligence system need not be a well-organized and unified activity in any sense. For example, in the system that supports a U.S. Army corps in Europe, a NATO army group commander generates information requirements that the corps commander clarifies for his corps sector and then conveys to the intelligence staff in the corps tactical operations center (CTOC). That staff and its Military Intelligence (MI) brigade support element and operational battalions translate these information requirements into a specific collection plan and, for some collectors, into detailed technical data on what must be collected. For collection, the corps staff relies heavily on the organic assets in its supporting MI brigade, but it also sends requests to collection activities owned by the U.S. Army at echelons above and below corps, the U.S. Air Force, and NATO allies, particularly the Germans and British. With various degrees of processing completed, information from these sources returns to the CTOC for integration into usable intelligence reports. The CTOC support element then distributes these reports, at various levels of classification, to U.S. Army and Air Force operators, non-U.S. operators, and NATO headquarters. U.S. Army and Air Force, non-U.S., and NATO communication systems support these flows of information. Some of these communications systems are dedicated to intelligence traffic, others are common carriers. These arrangements differ in each U.S. corps. Markedly different arrangements exist in non-U.S. corps.

When we speak of a system, then, we need not speak of a well-integrated activity controlled and optimized by a single owner. On the contrary, an intelligence system is a complex net of activities whose performance depends not just on technological factors but also on alliance, interservice, and intraservice politics, and on a wide range of human activities within the systems that constitute the intelligence system as a whole. Such a system is in continual flux as actors within its parts attempt to accommodate their behavior to the potentials offered by changes in personnel experience, technical and procedural innovations introduced, and new bargains struck throughout the system.

HOW AN INTELLIGENCE SYSTEM AFFECTS THE EFFICACY OF DEEP FIRES

To understand how an intelligence system supports the execution of deep battle, let us examine how it supports the Army's doctrinal three-step process for employing the Army's principal deep-battle weapon, the ATACMS. To use the ATACMS against deep-battle targets, the Army plans to "Decide, Detect, and Deliver." It first decides on what targets to strike. It then detects the location of these targets with enough precision to attack them with high confidence. Then it delivers the ATACMS against these targets. An intelligence system plays a central role in the first two steps.

An intelligence system assesses the situation on the battlefield as a whole, providing information on the current location, capabilities, and activities of Red units and on their likely intent and future actions. This allows the Blue commander to determine how specific Blue actions in the deep battlefield can support his total battle plan. He can focus his attention in the deep battlefield on specific Red units and actions and on specific locations and times and Red activities in these locations and time periods. He can then decide which Red units and locations to strike with deep fires. Targeting cells, which typically lie within intelligence assessment organizations, can then translate the commander's general guidance on targeting into specific targets.

Once targets are chosen, intelligence organizations can use existing information and manage collection of additional information to refine information on the likely location of these targets when strike assets are available to hit them. For deep fires by manned aircraft, target cells may specify locations and times for strikes as much as a day in advance. With new collectors that the Army expects to provide target-quality information on current location in near-real-time, the Army may be able to commit the ATACMS against targets on only a few minutes notice. Either way, intelligence assets process information collected on the deep battlefield into a form that allows deep fires at fairly precise locations and times. The Army can then execute the "detect" step of its three-step doctrine.

Although intelligence organizations play no direct role in the final "deliver" step, they can help assess battle damage assessment following an attack. Further, delivery is more likely to be successful the more accurate the location information and the faster this information is provided in the "detect" step.

To understand how well an intelligence system supports deep battle, we must have information on how well it can perform situation assessment, targeting, and battle damage assessment. Blue intelligence must develop a Red order of battle and use information from it to infer the enemy's intent and future actions. In practice, these are not separable tasks. An intelligence system relies on its inferences about Red intent to organize the mass of data it receives and from it refine a coherent order of battle. A coherent Red order of battle is necessary for confidence in inferences about Red intent. In practice, a Blue processing organization continuously simultaneously updates both the Red order of battle and inferences about Red intents.[1]

Information about both the Red order of battle and inferences about Red intent is important to situation assessment, targeting, and battle damage assessment. The reasons should be apparent for situation assessment. For targeting and battle damage assessment, empirical information about the location, identity, and type of a unit—information important to targeting—can always be refined with additional processing; this additional refinement will reflect Blue assumptions about Red intent. Further, any requirement to project Red activity into the future requires targeteers to move beyond the Red order of battle, necessitating assumptions about Red intent. The use of collectors that can deliver high-quality information on location in

[1] For a discussion of the dynamic relationship between basic intelligence data, like those in an order of battle, and higher level inferences about the total battlefield situation, see Kahan, Worley, and Stasz, 1989.

near-real-time reduces the importance of inferences about Red intent, but in almost all cases, they do not eliminate them.

Ideally, then, any analysis of how well an intelligence system works to support deep fires should consider its ability to assess the situation on the battlefield, to pick and locate targets, and to assess success against targets. These things will depend on its ability to maintain an accurate Red order of battle and high-level inferences about Red intent. It is much easier to model how an intelligence system develops and maintains an order of battle than to model how it makes higher level inferences. For example, it is easy to define a Red order of battle in terms of a simple list of unit attributes:[2]

- name (for example, "21st Guards")
- type ("motorized infantry," "command post," or "SA-12")
- echelon ("battalion" or "battery")
- location (UTM or lat-long)
- direction ("east," "west")
- speed (km/h)
- combat effectiveness ("40 percent," "90 percent")
- activity ("tactical assembly," "in march," or "in hide position").

No similar list suggests what it means to determine the Red commander's intent; predictable examples might include:

- Where is the enemy's center of gravity?
- What is the enemy's principal axis of approach?
- Where and when will the enemy commit its reserves or operational maneuver groups (OMGs)?
- What is the enemy's principal objective?

But many other questions can be important; and, to be meaningful, the phrasing of most questions about intent must reflect the context of an engagement.[3] Further, a variety of analytic efforts have been made to model how an intelligence system develops information on several of the attributes above. Analysts have not been successful in the far more difficult task of modeling intent.

As a result, we focus our attention on using the quality of Blue information about the Red order of battle to measure the performance of intelligence systems in the deep battlefield. We recognize that this is a partial measure, but we believe it is worthwhile to examine where the analytic basis for evaluation is the strongest. Users should keep

[2] U.S. Army, 1987; hereafter FM 34-1.

[3] Kahan, Worley, and Stasz, 1989.

this in mind when applying this approach in settings where it might affect the results.

SPECIAL PROBLEMS OF MODELING INTELLIGENCE DEVELOPMENT IN CENTAG

In U.S. Army doctrine and practice, the corps plays the central role in developing combat intelligence on the battlefield situation and targets. It controls substantial collection and processing assets and coordinates the development of an order of battle for the entire corps sector.[4] Hence, it is natural to direct any modeling effort to the corps. First, however, it is important that a focus on the corps will promote our general interest in the U.S. Army's use of the ATACMS to pursue deep battle in Central Europe. A review of the corps' role in developing intelligence relevant to the deep battle in NATO's Central Army Group (CENTAG)[5] will begin with a discussion of some basic aspects of intelligence development in CENTAG and their implications for modeling.

Intelligence Development to Support Deep Battle

Two special aspects of deep battle are important in CENTAG. The first is the question of who has primary responsibility for maintaining intelligence on the portion of the battlefield relevant to the ATACMS. The second is who has collection and processing capabilities to develop such intelligence. Those responsible for maintaining intelligence do not always have their own capability to develop it.

CENTAG oversees the development of intelligence in its entire area of interest. To do this, however, it depends on its subordinate commands to develop intelligence within their "areas of responsibility." Divisions are responsible for developing intelligence on the area within the Fire Support Control Line (FSCL). They report an aggregated version, typically an order of battle at the regiment/brigade level,[6] to their

[4] The organization and operation of a corps are not nearly so standardized as the organization and operation of units at lower echelons are. Nonetheless, the Army does provide a useful description of typical military intelligence activities at the corps level in FM 34-1.

[5] In wartime, the two U.S. corps currently stationed in Europe, V and VII Corps, are constituent commands within CENTAG. In the event of war, U.S. III Corps would provide a reserve for NATO's Northern Army Group (NORTHAG). Although much of our discussion could easily apply to either army group, our discussion of specific institutional arrangements concentrates on CENTAG.

[6] When developing an order of battle, each echelon typically maintains detail at two echelons below its own level.

superior corps. The corps have an area of responsibility that extends to the Reconnaissance Interdiction Planning Line (RIPL). They take in division information and add their own intelligence on the area between the FSCL and RIPL, aggregate their order of battle to the division level, and send it to CENTAG. CENTAG takes responsibility for developing intelligence on the area beyond the RIPL. In sum, intelligence at the CENTAG level is a patchwork of information from subordinate commands and information it develops on the area beyond the responsibility of those commands. An analogous situation exists at each subordinate command.

The portion of the battlefield of greatest concern to this study extends from the deep portion of the division area of responsibility to a point beyond the RIPL. That is, deep battle planners give increasing attention to operations forward of the FSCL. The ATACMS will also be effective on targets beyond the FSCL in a large portion of the corps area of responsibility. To choose targets, Army commanders need situation assessment information on portions of the battlefield that extend to an area beyond the RIPL. Technically speaking, then, divisions, corps, and CENTAG itself have responsibility for developing intelligence on portions of the deep battlefield. CENTAG is primarily concerned with situation assessment.[7] Corps and divisions must consider situation assessment and targeting information.[8]

Three different kinds of problems arise when we move from this division of responsibility to the development of combat intelligence. First, CENTAG has few organic assets to use to develop intelligence. It relies primarily on U.S. and German assets, which are beyond its direct tasking control, to collect and process information on the portion of the battlefield beyond the RIPL. As a result, CENTAG is a consumer, not a producer, of whatever intelligence is available in this portion of the battlefield. CENTAG also relies on the national assets of its subordinate commands for information inside the RIPL. This raises the second problem. The technical collection and processing capabilities of the U.S. corps exceed those of their neighboring non-U.S. corps and procedures, and facilities do not exist to allow rapid

[7] CENTAG, in coordination with the 4th Allied Tactical Air Force (4ATAF), chooses targets for aircraft at fixed locations on the deep battlefield. For the ATACMS, however, CENTAG's primary intelligence concern would be situation assessment.

[8] Whether CENTAG or the corps will control the employment of ATACMS has not been settled. At this time, CENTAG and 4ATAF will probably plan the deep battle, and the corps will execute the Army's part of it. CENTAG will specify goals and the corps will operationally control the ATACMS, translate CENTAG goals into specific missions, and execute the missions. Our discussion assumes this distribution of responsibility. Either way, however, CENTAG remains responsible for maintaining a situation assessment beyond the corps area.

communication of intelligence information between corps. Hence, the quality of intelligence is likely to change as we cross corps boundaries within the RIPL. Finally, the U.S. Air Force has collection assets that can make a major contribution to intelligence on all parts of CENTAG's deep battle area. Difficulties exist in getting requests for U.S. Air Force data from the U.S. Army and other nations' forces to the U.S. Air Force and in getting data back in a timely manner. Coalition and joint planning have created a complex net of formal and informal channels for moving intelligence information and requests for information within CENTAG and a strong preference within each U.S. Army command for focusing its intelligence development on information from organic assets.[9]

The All-Source Analysis System/Enemy Situation Correlation Element (ASAS/ENSCE),[10] scheduled for full introduction into the European theater during the next decade, will address only some of these problems. Many expect ASAS/ENSCE to make dramatic improvements in the speed of developing and distributing intelligence products and in their precision. This may well be true within U.S. corps sectors. ASAS/ENSCE continues the Army's emphasis on the corps as the central player in intelligence development. It will also improve coordination between the U.S. Army and the U.S. Air Force. But it is not designed to develop or manage information on areas beyond the U.S. corps area of responsibility or to coordinate communication on non-U.S. corps sectors or NATO areas beyond the RIPL.[11] In fact, it is not designed to facilitate cross-corps communications that could be critical to deep battle.

Basic Institutional Factors Relevant to Modeling

These factors can complicate efforts to model effective communications or priorities determination within the CENTAG intelligence "sys-

[9] For a useful discussion of these issues, see Kahan, Worley, and Stasz, 1989.

[10] ASAS/ENSCE is a developmental system of hardware and software that is expected to automate many aspects of communication and data management for intelligence activities within a corps. The Army and Air Force are jointly developing this system through the Joint Tactical Fusion Program.

[11] Because ASAS/ENSCE is not a mature system, it is hard to predict either what it will look like in Europe or how well it will actually perform. In all likelihood, it will evolve over time and adapt to the European setting. For example, the U.S. Echelons-Above-Corps Intelligence Center (EACIC) has had an effort under way for several years to develop what has been called a European ASAS. It is primarily a communication and data management system designed to improve communication beyond the corps context. This system is not being closely coordinated with the development of ASAS/ENSCE in the United States.

tem" and the way these affect the quality of intelligence in different parts of CENTAG. To see this, consider the simplest aspect of modeling intelligence development within CENTAG—a single U.S. corps—and then look at what happens when we go beyond a corps sector within CENTAG.

To a first approximation, we can model one U.S. Army corps area of the deep battlefield without reference to non-U.S. or Air Force assets. The Army organizes its combat intelligence system in Europe around the corps. Each corps has all the collection and processing assets it needs to develop situation assessment and targeting intelligence on a large portion of its area of responsibility in the deep battlefield. Although Air Force aerial platforms, U.S. satellites, and German human intelligence (HUMINT) enhance this intelligence, each corps relies primarily on its own assets and can develop credible intelligence using only those assets. A standard U.S. corps intelligence system exists. U.S. corps are deliberately designed not to be standardized, and each U.S. corps with responsibilities in Europe develops combat intelligence on the deep battlefield in a different way, raising difficulties for the modeler even in this simple case.

Suppose now that we want to model situation assessment beyond the RIPL. Corps collection assets are not nearly so useful at this distance. CENTAG must rely heavily on German HUMINT and on technical intelligence from the EACIC and the U.S. Air Force, which use a variety of U.S. collection and processing assets to develop intelligence on the very deep battlefield. In formal protocols, if a U.S. corps wants information from the EACIC, it asks CENTAG for the information. Then through NATO channels, CENTAG determines whether the EACIC is the appropriate source and, if so, forwards the request to the EACIC. In fact, each U.S. corps is in constant contact with the EACIC, and informal requests and data flows are typical. German corps have only the formal channel of communication; they have similar informal ties to their own national sources, organized within their intelligence systems, that U.S. corps cannot exploit.

Suppose we are interested in modeling the use of a U.S. corps' ATACMS in another corps sector, which can substantially increase the usefulness of the ATACMS. However, intercorps intelligence development and communication are not as good as that within a corps. Because the corps represents the heart of the U.S. Army intelligence system and corps intelligence systems are not standardized, they are not well designed to accommodate communication and coordination across corps sectors. Even coordination between U.S. corps is difficult. For example, V and VII Corps maintain different situation assessments of common regions of the deep battlefield and make few attempts to coordinate them in peacetime or in exercises.[12] Coordination between U.S. and non-U.S. corps is even harder. Because German corps put less emphasis on deep battle than U.S. corps, they put fewer resources into assessing the deep battlefield and fewer still into developing targets there. They would simply not be prepared to provide the information required, in a timely way or in the detail required, to facilitate employment of ATACMS in a U.S. corps sector against targets in a German corps sector. This will be true whether decisions on the use of the ATACMS occur above the level of U.S. and German corps, at CENTAG, or at the U.S. corps itself.

These examples tell us that very different levels of quality can exist on different parts of the CENTAG deep battlefield because of differences in the resources available to develop intelligence and differences in access to these resources. Different levels of quality can persist for different users within CENTAG on any part of the battlefield as much from behavioral and political considerations as from engineering or technical aspects of intelligence development. Any model attempting to capture such differences must reflect a subtle institutional knowledge of intelligence development in CENTAG.[13]

Implications for Modeling

CENTAG is a complex setting with heterogeneous intelligence capabilities and priorities. For our purposes, we must understand intelligence development within a U.S. corps. The first point that a modeler must settle is how much institutional detail on intelligence activities beyond the U.S. corps is really necessary. If the primary interest is targeting, perhaps a model of a single corps area, with some allowances for Air Force and perhaps satellite assets, will suffice. For a full understanding of the situation assessment that underlies target choices, additional information on the EACIC, the collectors and processors it uses, and its formal and informal links to CENTAG and the U.S. corps is important. Shallow fires require emphasizing certain division collection and processing assets. Cross-corps fires need detail on differences between corps and their interactions. We need not develop two complete models for the corps involved, but the nature of inter-corps communication and some basic notion of their relative capabilities and priorities will be important.

[12] They do not share areas of responsibility. But they have both chosen to maintain their own orders of battle on portions of the battlefield beyond their areas of responsibility. CENTAG uses only the information that each corps develops for its area of responsibility.

[13] Although CENTAG has a particularly complex institutional setting, institutional factors should be important in other combat organizations as well.

Different parts of a study may examine each of these questions. In that case, we may want more than one model. Each would require a strong representation of intelligence development within a U.S. corps. Beyond that, considerable variation could be desirable. This would be easiest if a single underlying modeling approach could be used to develop them all. A flexible modeling system might use different configurations of collection, processing, and communications assets, coupled with differing priority systems, to represent these variations.

Intelligence systems that support the ATACMS in the future could be very different from those we observe today. We have already mentioned that ASAS/ENSCE is expected to change radically information management within corps. Other innovations, such as quick-fire channels and unmanned aerial vehicles (UAVs) carrying near-real-time collectors, could have a similar effect. So could reorganizations within CENTAG. The future holds great potential and a modeler should be prepared to consider many variations on that potential. A flexible model should be detailed enough to accommodate alternative assumptions about individual elements of the intelligence system and institutional factors that organize it.

This line of argument suggests that information on current intelligence development within CENTAG is not particularly useful. In fact, despite the best laid plans, important elements of the U.S. corps intelligence and communications systems in CENTAG date back to the Korean War. Changes over the next decade should be similarly evolutionary. Certainly, institutional difficulties like those discussed above will take a long time to ameliorate. Any model should be able to reflect our current understanding of CENTAG; it should also be able to reflect serious excursions from current practice. This reinforces the need for a flexible model that can be varied at different levels of detail.

THE KEY: EFFECTS OF INCREMENTAL CHANGES IN INTELLIGENCE SYSTEMS

How changes in particular parts of an intelligence system affect the quality of Blue information on the Red order of battle in the deep battlefield is the central question of our approach. For example, how would 24-hour availability of data from JSTARS in a corps sector affect the quality of information on unit location or unit identity in the corps' deep battlefield?[14] What about 12-hour, intermittent availability?

[14] JSTARS is the Joint [Army-Air Force] Surveillance and Target Acquisition Radar System, a radar mounted on a standoff, aerial platform with real-time data links to dispersed ground stations. It is currently not in the force. If it were added, it should

Quality can obviously change over the course of an engagement and differ for different types of units and in different parts of the battlefield. These differences may be important to policy decisions about the availability or use of collectors, processors, or communication lines. Hence, we want to preserve a richness of detail about how changes in an intelligence system affect the quality of information. By the same token, wε want to be able to aggregate across these differences, where they are not important, to generate simpler summary measures of information quality. Hence, we must measure information quality in comparable terms across attributes, unit types, locations on the battlefield, and so on.

To meet these specifications, we employ a carefully detailed, quantitative model of information quality in an intelligence system that allows us to vary the availability or use of specific elements of the system, one at a time. We posit an engagement scenario that describes how Red units behave on the deep battlefield over the course of an engagement. Our model simulates the quality of information produced by a baseline intelligence system in this scenario. We then perturb the intelligence system to simulate an incremental change in the availability or use of a key element. *Using the same combat scenario*, we generate measures of the quality of information that this new system produces. Differences in the quality of information generated by the two systems measure the effect of the perturbation. By repeating this sequence under varying assumptions about how the intelligence system works, or for varying combat scenarios, we can test the sensitivity of our results in the face of irreducible uncertainties about combat intelligence.

OTHER WAYS TO EVALUATE INTELLIGENCE SYSTEMS

The approach we present is only one of several that analysts have used to evaluate the performance of intelligence systems. Others have been used and we can compare them with the *Red order of battle* approach.

An *engineering* approach emphasizes the technological capabilities of individual elements of an intelligence system or simple combinations. For example, it might consider the resolution of a sensor or a sensor's ability to discern a particular object on the battlefield given its resolution. Similarly, it might consider the data flow rate, queuing, and

make a large contribution to the quality of information on location and almost none to the quality of information on unit identity; knowing that is fundamental to knowing the incremental value of JSTARS to an intelligence system.

delay time on a specific communication link or in a specific intelligence processor. In some cases, it might involve the performance of a set of components. For example, it could consider the compatibility of sensors, communication links, and processors in a "single thread" and their joint ability to convey to a central location a message about a specific thing on the battlefield. Such analyses are most useful in setting the technical specifications for individual components of an intelligence system or tuning these components to assure that they perform as well as possible together.

A *connectivity* approach views an intelligence system as a network that passes discrete messages from place to place. Nodes typically represent sensors or processing activities; arcs typically represent generalized communication links between these nodes. Nodes do not alter messages in any way, and representations of such nodes convey no information about physical or human assets that might be present at a sensor or processor site. Delays need not even occur in passing messages through nodes. Arcs may convey information on alternative links between two nodes, but they typically do not reflect detailed engineering data on every link. Such a network depiction allows analysts to examine how messages move from one node to another. It is most useful in studying how fast information can move and how robust information flow is in the face of combat damage or reliability considerations.

A *combat outcomes* approach considers intelligence activities from an entirely different perspective. If the two approaches above emphasize inputs to an intelligence system, this approach considers the ultimate output of an intelligence system—how it affects combat outcomes. It includes an intelligence module as one of many components in a combat simulation. It then alters parameter values or rules within the intelligence module and observes how these changes affect the final outcome of simulated combat. The intelligence module can be very detailed or fairly simple.[15] This approach is most useful when comparing the efficacy of investments in intelligence and nonintelligence combat capabilities that both contribute to total combat effectiveness. It is also useful in studying intelligence activities when changes in them affect combat outcomes enough so that we cannot understand their total effectiveness without studying how they affect combat. For example, increased use of airborne collection platforms early in a conflict could lead to heavy attrition of the platforms that cripples the intelligence system later; the opportunity cost of focusing intelligence

[15] Two very different approaches are offered in Gamble et al., 1987, who offer a very detailed approach that ultimately requires some human intervention to execute properly.

assets deep rather than shallow could be small if deep intelligence allows such heavy attrition or delay of Red units in the deep battlefield that the close battle becomes considerably more manageable.

Our *Red order of battle* approach builds on the engineering and connectivity approaches and could enhance the combat outcomes approach. It emphasizes a particular product of the intelligence system that contributes substantially to a commander's combat effectiveness. That is, it asks how well the Blue intelligence system depicts the Red order of battle and how long it takes for the Blue commander and his operations staff to become aware of important changes in this order of battle.

This approach draws on information from engineering and connectivity studies but does not incorporate as much detail as those others do. It sacrifices details on specific portions of the intelligence system to gain better understanding of how the system as a whole works. It is also easier to isolate aspects of an intelligence system's performance that a commander believes are important than using a combat outcomes approach would be. It is often difficult to determine the precise channels through which changes in a single combat capability affect ultimate combat outcomes, particularly when the combat simulation is complex, the algorithms and rules in it are not transparent, and outcomes of the simulation are sensitive to aspects of the scenario that are important to the combat capability in question. Focusing on a system's ability to perceive the Red order of battle gives a commander direct information on the performance of one combat capability—intelligence—that he can interpret without having to parse the details of a combat simulation and decide which parts of it he believes under what circumstances.

The order of battle approach need not replace the other approaches that evaluators have used. It complements them by providing additional insights about the effects of changing an intelligence system. For example, suppose we are interested in the effects of introducing a new imagery intelligence (IMINT) system like JSTARS. Engineering studies are required to choose and configure antennae, analyst positions, communication protocols, and so on. Connectivity studies can inform us about what users benefit from JSTARS data and how fast they get information based on these data. A combat outcomes analysis can compare the desirability of JSTARS with that of other assets like attack helicopters or missiles that the United States might buy. The Red order of battle approach can ask how JSTARS contributes to situation assessment, target acquisition, and the execution of targeting plans. It asks how the addition of JSTARS affects needs elsewhere in the intelligence system and suggests specific activities in that system that JSTARS might allow to be replaced without jeopardizing the maintenance of important intelligence products. Good answers to questions like these can easily improve the execution of the other approaches by improving their depiction of the intelligence system as a whole and how it might change when one part of it changes.

COORDINATION WITH VIC

In executing our approach, we rely on Army models and capabilities. One way we do this is to make the fullest use possible of the Army's Vector-in-Commander (VIC) corps combat model.[16] The Army has chosen VIC from among several models to be the official corps model it will rely on for scenario development, combat modeling, and other policy analysis.

Using VIC offers several benefits. First, the TRADOC Analysis Command (TRAC) has used VIC to develop some authoritative Central European combat scenarios for U.S. corps. The Army currently uses these models in its force development activities; VIC allows us to use these scenarios as well; we therefore do not need to develop scenarios and we use assumptions consistent with those that underlie Army planning. Second, VIC generates a detailed account of how Red forces behave in the deep battlefield that we can use as a basis for empirical input to the Blue intelligence system. It is the only corps model we have seen that provides the depth of detail on Red status that we believe is needed to model Blue intelligence assessments of Red status.[17] VIC also provides a method for modeling Blue's use of collectors and their information, and it embodies standard Army assumptions about the performance of these sensors; we have found that these assumptions provide a useful baseline for our own analysis.

We have carefully tailored our approach to exploit VIC wherever possible, although our approach is not totally dependent on VIC. Without changing its basic structure or logic, our approach could be tailored to use information from alternative combat models that generate information on Red activity and Blue's collection efforts against this activity. The Army will undoubtedly consider new corps models in the future, and our approach could be tailored to these new models without serious difficulty.

[16] VIC combines the Vector Research Corporation ground combat model with the Commander air model to represent the full range of AirLand Battle functions in the context of a U.S. corps battle. For documentation, see Gamble et al., 1987.

[17] It provides continuous information, through the course of an engagement, on all of the attributes listed above for each Red unit. It also models every Red unit type likely to offer a good target for the ATACMS.

SUMMARY

To support its efforts in the deep battlefield, the U.S. Army needs good information on the general battlefield situation and the location of targets for deep fires. It uses a complex system of intelligence systems to generate this information. These include collectors, processors, and communication lines. *We seek a method to evaluate the effects on potential performance of incremental changes in combinations of these systems.*

We focus on the intelligence system in a U.S. Army corps. The corps lies at the heart of the Army intelligence system, and any view of how the Army might use the ATACMS in Central Europe emphasizes the importance of the corps intelligence system to support that use. A complete intelligence system, however, must consider elements outside a corps. Our approach allows a wide range of variations on a corps system within a common modeling environment. The simplified "corps" system we present here to illustrate our approach includes a joint system, the JSTARS, that in fact lies beyond the corps' complete control. We could just as easily add other elements in a similar way. More generally, the approach is extremely flexible and can model configurations of corps intelligence systems and many extensions beyond them.

Our approach uses the quality of a Blue intelligence system's information on the Red order of battle in the deep battlefield to measure the performance of the Blue intelligence system. It allows us to vary the availability and use of elements of an intelligence system one at a time and simulate the effects of these variations on the quality of Blue information about the Red order of battle. By comparing the quality of information generated by intelligence systems with different configurations, we can measure the effects of their differences. We can also determine how sensitive differences in information quality are to variations in assumptions about how intelligence systems work or to circumstances on the deep battlefield.

Other methods can be used to evaluate intelligence systems. Engineering methods tend to examine individual components of an intelligence system and judge them in terms of technical criteria, such as resolution. Connectivity methods focus on how the elements of intelligence systems are linked together and ask how fast discrete messages can flow through them. Combat outcome methods view intelligence systems as one component in a total force and ask how changes in intelligence activities affect the final outcome of an engagement. *Our approach complements these alternatives, drawing on engineering and connectivity studies for its inputs and offering more detailed results*

on intelligence quality per se than combat outcome methods offer. Each approach is designed to answer different kinds of questions.

We rely as much as possible on existing Army models and capabilities mainly by using *the Army VIC corps combat model to simulate the behavior of Red units in the deep battlefield and to simulate collection of primary intelligence on this behavior.* VIC allows us to use authorized Army scenarios and facilitates our efforts to incorporate Army assumptions about combat intelligence into our analysis. We could use other simulations of Red unit behavior and Blue intelligence collection to implement our approach.

III. MODELING INFORMATION FLOWS IN AN INTELLIGENCE SYSTEM

Broadly speaking, an intelligence system gathers diverse information on specific aspects of what Red is doing on the battlefield and uses this information to infer a more complete picture. Both the information and the processes that Blue intelligence uses to do this are complex. Blue intelligence effectively must split this job up and move information from one activity to another in an attempt to improve its information on Red behavior. We want to understand how changes in elements of a Blue intelligence system change its activities and how these changes affect the quality of information that it produces. For analytic purposes, we can split our inquiry into two closely related questions. First, how should we model information flows through a Blue intelligence system and the factors that affect this information flow? And second, as information flows through, how should we model changes and factors affecting changes in its quality?

A CONCEPTUAL VIEW OF INTELLIGENCE DEVELOPMENT

For our purposes, an intelligence system includes collectors, processors, and communication systems. The process that such a system uses to develop intelligence can be quite complex. But it is possible to capture the essence of intelligence development in such a system in fairly simple terms. To see how, let us review first how information flows from place to place in an intelligence system and then how information flows over time in a system.

The Structure of Information Flows

Figure 1 uses a simple "single-thread" system to illustrate the key information flows in an intelligence system. Information flows from the battlefield to a final user through a single channel of communication. Letters in parentheses identify key steps in this flow.

Step A. The intelligence manager distributes information on collection and processing priorities, which reflect the goals and priorities of the user, presumably the commander. For our purposes, however, the manager is the ultimate source of all information in the system on

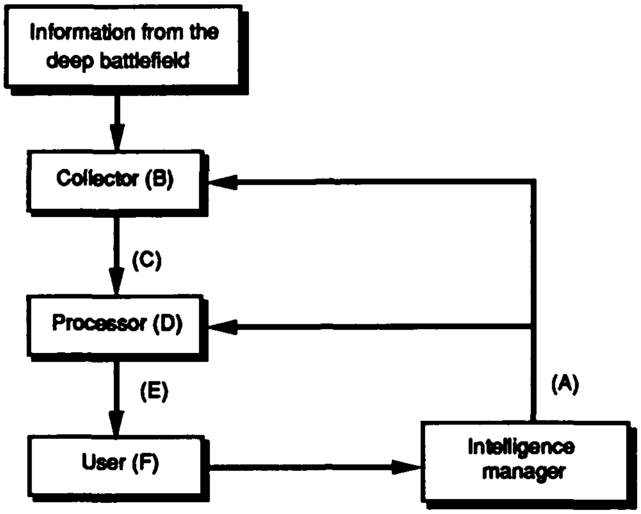


Fig. 1—Basic data flows in intelligence development

priorities. In particular, the intelligence manager tells the collector where and when to deploy in the future and what to look for. It tells the processor what is important and how to commit its resources when transforming new information into new conclusions. *Step B.* The collector gathers information flows from the battlefield and filters this information. The collector places a portion of it in the communications system to convey it to its processor. *Step C.* On the basis of instructions from the collector, based in turn on priorities set by the intelligence manager, the communications system conveys the most important, "high-priority" information to the processor first. *Step D.* The processor update its files, draws conclusions, and places relevant conclusions in the communication system to convey to its user. Again, it can process the most important information first and most thoroughly, convey conclusions based on this first, and instruct the communications system to give high priority to important messages to the user. *Step E.* The communications system sends relevant conclusions to the user in accordance with instructions from the processor. *Step F.* The user determines what additional information it requires and sends feedback to the intelligence manager. The intelligence manager translates the user's information needs into instructions for the collector and processor to start over again.

The most striking aspect of this system is that, during any step, information flows in only one direction between any two elements of the system. In practice, processors will query collectors directly for clarifications; similarly, users will query processors directly. Our representation implicitly views such queries as microprocesses below the level of concern. They simply facilitate the development and flow of information in one dominant direction between elements of the system.[1] Although they can be critical to assuring the integrity and efficiency of the system as a whole, we do not have to understand them in detail to understand how new information from the battlefield progresses through an intelligence system, updating its databases, and how users convey their new requirements back to collectors and processors.

In this view, then, an intelligence system moves information in a closed loop. It receives priorities from a user, uses these priorities to collect and process information, and conveys conclusions based on these priorities to the user, who then initiates a new cycle. This basic cycle lies at the core of any combat intelligence system, no matter how complex. Our model is a slight variation on it. It is still important to understand other aspects of intelligence development; but we can model any combat intelligence system by placing one or more cycles like this at the heart of the model.

When the intelligence system is somewhat more complex, consider a system with three collectors, three processors, two users, and more than one intelligence manager. Figure 2 illustrates the information flows in this system.

Step A. In a more complex system, we can expect more than one intelligence manager. In Central Europe, for example, the Blue intelligence system includes U.S. and non-U.S., Army, Air Force, and "national" intelligence managers. They attempt to coordinate priorities, but in the end each sets specific priorities for the assets it oversees.

Step B. Now three collectors receive and filter information on the deep battlefield. They will typically observe different aspects of it at different times. For example, one may monitor radio communications (communications intelligence or COMINT). The next might gather information on Red radars an hour later (electronic intelligence or ELINT). The third might use radar to detect and measure the movement of Red vehicles on the deep battlefield at yet another time (one form of imagery intelligence or IMINT). In each case, however, a collector uses collection priorities to determine where to look, when to

[1] Cf. Kahan, Worley and Stasz, 1989.

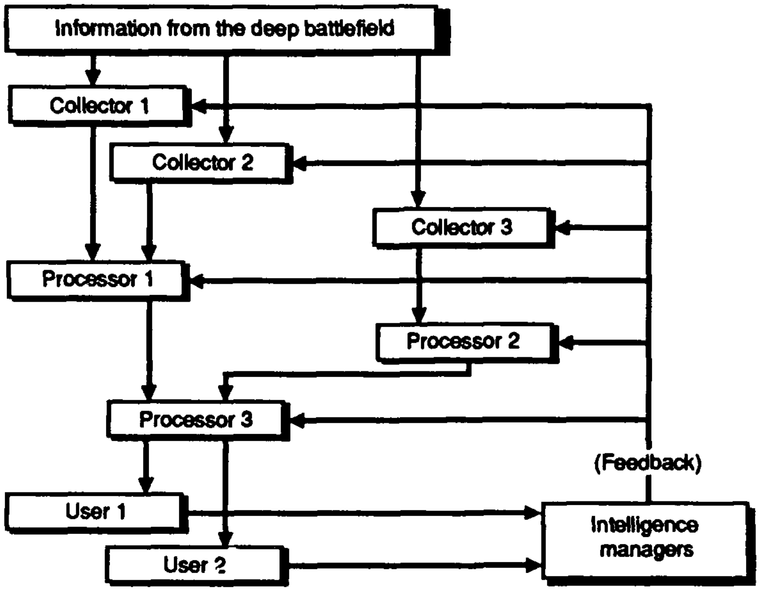


Fig. 2—A more realistic set of data flows in intelligence development

look, and what to look for and filters information before conveying it to a processor, often with information about its priority.

Step C. The communication system accepts instructions about priorities from collectors and sends their information to processors.

Step D. The processors integrate the information from these collectors with information in their databases and with one another. They attempt to construct a coherent picture from information on different aspects of the battlefield collected at different times. In Fig. 2, two collectors feed information to one processor while the third collector feeds a dedicated processor of its own. Each of these processors accepts information, processes it into updated information in its database, and develops conclusions to the third processor. They convey information on priorities to the communication system that carries these conclusions.

The final processor in Fig. 2 uses new information from other processors to update its database and to develop conclusions for its various users. Different users may require information on different parts of the battlefield, on only targets or only situation assessment, at different frequency rates and levels of aggregation, and so on. The final processor sorts through these information requirements and places relevant conclusions, with information on their priority, in the communication system to convey them to users.

This more complex system differs from the single-thread system in important ways. Most important, processing occurs in several places. The system processes information from collectors, develops interim intelligence products, places them in the communication system to other processors, develops more refined interim intelligence products, places them in the communication system, and so on. As a result, several communication steps can reside within this "processing" step. Some processors must accept new data from more than one source and send information to more than one destination. Again, this intelligence system allows communication in only one direction between processors. As in the single-thread case, queries move in both directions between any two elements of the system and are important to facilitating the operation of the system. But basic, intermediate, and final inferences about the deep battlefield relevant to users typically move in only one direction between any two elements, the direction shown in the figure.

Step E. The communication system accepts instructions about priorities from processors and sends their information to users.

Step F. Users send updated information on their requirements and priorities to intelligence managers.

Figures 1 and 2 depict closed loop systems in which information flows in a dominant direction between any two elements and in a cycle when all elements are considered. The single-thread system contains one loop; the more complex system includes many interacting loops. Any new sighting on the deep battlefield potentially initiates a flow of information through a collector, one or more processors, to one or more users who then adjust their needs in a way that affects later sightings, using the collector that initiated this cycle or some other collector; and processing relevant to the information collected. Our model uses a variation on a simple loop like that in Fig. 1 as the basis for a model of an intelligence system with information flows like those in Fig. 2.

Information Flows over Time

Time is important to intelligence development for two reasons. First, the environment of an intelligence system changes through the course of an engagement, and *when* flows occur affects *how* they occur. Second, the information flows above do not occur instantaneously. Several factors make these flows occur over a period of time.

Over the course of a dynamic combat engagement, what Blue intelligence can see on the deep battlefield changes and what the Blue commander wants to see on the deep battlefield changes. The first point means that if collection occurs at different times, it will observe different events on the deep battlefield. The second means that the priorities users set for collection and processing will change over the course of the engagement, and those will change the priorities they convey to the communication system. If the intelligence system works well, it should actually influence the course of an engagement. In a fully developed model of intelligence development, we would want what Blue sees on the deep battlefield to depend on past actions of Blue intelligence. Our model stops short of this)ast feature.

Intelligence activities in a dynamic environment are complicated by the fact that intelligence development takes time. Each of the steps above takes time. Good empirical data are typically not available to say how much time each step takes. And within the intelligence community subjective judgments about times differ substantially, although everyone agrees that how much time intelligence development takes is a critical part of any understanding of an intelligence system. Consider the factors that contribute to delay in each of the above steps.

Step A. It takes time for the commander's staff to review its situation and goals and revise its priorities. Under current procedures, the intelligence system receives and communicates most information on priorities on a regular schedule; planned communication times facilitate coordination in complex organizations. One effect of this system is that time can pass from when the commander revises his priorities to when his staff communicates these to collectors and processors.

Step B. Collection takes time to plan and execute. In fact, collection managers usually plan their use of assets several days in advance and report their schedule to processors so they can know when to expect certain kinds of information. Again, regularity promotes coordination even as it limits flexibility and responsiveness. Managers change this schedule only in response to very high priority requests. They can change collection priorities within a schedule more easily, but even this requires lead time to prepare collection software and so on. Execution also takes time. Airborne collectors, the principal source of information on the deep battlefield, take time to reach station, collect data, filter and approve preliminary intelligence products, and communicate them to a ground station that can place them in the communication system.

Steps C and E. Today, the largest source of delay in U.S. Army intelligence systems occurs in the communication systems they use. The systems typically convey information at a slow rate and are often not reliable. Intelligence activities often send messages on more than one channel to increase the probability and speed with which they arrive at their intended destinations. On dedicated communication lines, intelligence messages can compete with one another, slowing arrival times as loads rise. On common communication lines, intelligence messages compete with other messages and suffer slower arrival times as loads rise on the communication system as a whole. The communication system uses priority levels to sort messages and place higher priority messages on faster, more reliable channels.

Step D. Processing takes time. In some cases, processing occurs on a regular schedule. Hence, a processor may not deal with available data until the prescribed time arrives, even if resources are available. In some cases, processors do not deal with new data until enough accumulate to constitute a new batch. And once processing starts, it takes time. Automated systems can produce interim intelligence products in fractions of a second; human systems can take hours. When processing resources are strained, new data queue, adding further delays. Given the resources available, a processor can enhance the quality of one piece of information at the expense of another by shifting resources between the two. Information on priorities can encourage allocation of resources to high-priority information. Applying more resources to a piece of information, of course, may enhance its quality, but it may also increase the time required for processing; the processor must continually make tradeoffs between the use of additional resources and delays that result to ensure the best use. Priorities can also override standard schedules and batch practices.

Step F. It takes time for a commander to assimilate new information. It is probably not possible to state objectively when a user benefits from a typical new piece of intelligence. For simplicity, we treat such delays as beyond our purview. A user has new intelligence when he receives it; the time of his receipt closes each cycle.

When all of these delays are considered, the time required to execute these six steps, including communication time implicit within the processing step, can easily exceed 24 hours; commanders can expect to plan their actions on the basis of information that is many hours old. When the situation is changing on the battlefield, such delays greatly degrade the commander's confidence in his understanding of the situation, making accurate targeting extremely difficult.

Implications for Modeling

We can make a complex intelligence system analytically tractable by modeling it in the right way. Our real interest is in asking how the availability or use of a collection, processing, or communication asset affects the quality of information available to users. To answer this question, we need to reflect the following factors in our model. (1) Any asset whose effect we wish to model must be included as an entity in the model. For example, JSTARS could become a "collector." Its Ground Support Module, which maintains and updates databases based on JSTARS data, could become a dedicated "processor." (2) We want to build up information about the system as a whole from information on its parts. We do not need detailed information about each part; we need the minimum information required to relate each asset included to the larger system. (3) Information flows systematically in an intelligence system. To understand how any individual element affects information quality, we need to understand how it affects the flow of information. (4) We need to know how any part of the system affects the time required for the whole system to develop information. (5) A major instrument available to users to affect the performance of the system is control of priorities. We must be able to show how the system responds to different priorities. (6) The performance of an intelligence system will depend on the combat scenario in which we judge it.

KEY ELEMENTS OF OUR MODEL AND THEIR POLICY RELEVANCE

Network Representation

It is natural to think of the information flows in Fig. 2 in terms of a network model. Collectors, processors, and users constitute the nodes in the network. Communication links constitute the arcs that link these nodes. This is our approach. Any collector, processor, or user can be represented as a node that receives information along arcs from one or more nodes and sends information along arcs to zero, one, or more nodes. We initiate the flow of information by placing new information in the collection nodes when collection occurs on the battlefield. This is the only place that new information can enter. The model then simulates the flow of this information through the network. The information that enters a node need not be the same as what leaves; by definition, processing tends to change the information that moves through the network. Hence, we are not simulating simple connectivity; but we use a network like those in connectivity studies as a framework for our own analysis.

Delays in Processing and Communication

Communication and processing delays slow this information flow. In the model, it takes time for information entering an arc to cross it. It also takes time for information entering a processor node to induce new information to leave the processor. We are less concerned with what determines the size of each delay than with how delays affect the performance of the system as a whole. As a result, at any point during a simulation, we set delay times for each node and arc as simple constants that depend only on the priority of the information passing through. These constants can take different values over the course of a simulation; their values cannot respond to outcomes from the simulation.

Messages in the Network

To understand the mechanics of the model, it is useful to think of each information flow across an arc or through a node as a "message." A collector receives a discrete "message" from the battlefield. Following filtering, it then sends a discrete "message" to a processor. This processor may then send zero, one, or more discrete "messages" to another processor or a user. This is in fact what occurs in the model itself. But these "messages" do not correspond to actual battlefield messages. That is, we are not conducting a connectivity study, which might look at the number of actual messages generated by an intelligence system, the number of bits associated with each message, the bit rate of each communication arc, and the queuing that results from actual message flow in the intelligence system. We are instead looking at messages that move information through nodes or from node to node via an arc.

These messages push something like an information quantum through the intelligence system. They process battlefield information to enhance it and convey the content of new battlefield information or new processing from one place to another in the network. In effect, they allow one node to tell another, "Our image of the battlefield has changed and we think you should know how." A message that conveys information along an arc from one node to another might correspond to many real messages required to achieve this transfer. For example, one node could send a preliminary message that the receiving node routinely ignores. Or the receiving node might require two messages about a change before it responds by accepting the information; our model would represent this transfer in terms of one message and portray the delay between real-life messages as processing delay in the receiving node. Similarly, a receiving node may typically back-brief what it has received before accepting it.[2] The exchange associated with a back-brief could generate several messages in both directions; our model would include only one message that moves a quantum of information in one direction.

Our messages are also much more focused than messages in an actual intelligence system. In our model, each message signals a change or transfer of information about an individual Red unit-attribute. We define information on a unit-attribute as a statement about one of the following attributes of a specific Red unit: name, type, echelon, location, direction, speed, effectiveness, or activity. These are the basic attributes relevant to the Red order of battle. A single real message in an intelligence system would typically concern information relevant to several of these on several Red units. In fact, the real message traffic in an intelligence system typically concerns much greater detail. It may count trucks, identify technical characteristics of radars, or indicate a speaker's nationality. Although such detail is obviously critical to the real operation of an intelligence system—in a sense, the manipulation of such detail *is* intelligence development—it lies beneath the level of our analytic concern.

In sum, the messages that move information around our model really bear little practical relationship to the messages in a real intelligence system. Collectively, they convey the same information that messages on a real system convey. But their size, content, and number differ radically.

Priorities

The discussion above considers two kinds of priorities. The first are those associated with the timing and location of collection. The second are those associated with types of information that collectors and processors should always give special attention.

We reflect the first kind of priority in terms of a collection schedule. It defines the specific location and timing of each collector mission in terms that VIC can accept.[3] These include the exact orbit that a collector would fly and the exact times when that orbit would start and finish. Data that can be placed in VIC also indicate what each collector can see on such an orbit. They typically indicate a swath of Red

[2] Component A back-briefs component B on information that B has given A by restating the information in a form more suitable to A's use and checking with B to ensure that this interpretation is compatible with the information that B provided.

[3] For details, see Gamble et al., 1987.

territory parallel to the orbit and bands within the swath (also parallel to the orbit) showing what percentage of a unit located in that band a collector can see from the orbit and uses to collect information. The collection schedule reflects the commander's priorities and resource constraints on the collection system. We would use a similar approach if we were to coordinate the model with combat simulations other than VIC.

We reflect the second kind of priority in terms of specific Red unit-attributes. The *form* that these take in our model differs from the priorities one might find in a commander's Priority Information Requirements (PIRs) or his targeting priorities; but they are meant to embody the same information about priorities that a commander conveys with these formal instruments. For example, if the commander determined that the speed and direction of movement of the 5th Guards Army was a priority, our model would label the speed and direction of movement of each unit within the 5th Guards Army as a high-priority information item. If the commander wanted the location of all major command posts, our model would label the location of each major command post as a high-priority information item. Such priorities typically change in a regular daily cycle during combat; our model reflects this by allowing the priority placed on each unit-attribute to change over the course of a simulated engagement. The values of priorities cannot respond to outcomes from the simulation.

Feedback Versus Departures from a Baseline

The discussion above raises several opportunities for feedback in the model. (1) The performance of the intelligence system can influence the course of an engagement; hence, the intelligence activities that occur today should affect the Red activity that Blue observes tomorrow. (2) Even if this were not true, the quality of information developed today should affect users' incremental demands for information and hence the priorities that they set for information development tomorrow. (3) The amount of intelligence traffic moving on communication lines should affect the total loads on these lines and hence the delay times on them. (4) Similarly, the amount of processing demanded in a processor should affect its load and hence the delay time associated with processing. More subtly, changes in delay times among processors could lead managers to shift loads among processors.

Modeling such feedback can be demanding. Even the simplest feedback—probably that associated with communication loads—would require repeated computation of a system-wide equilibrium in which the delay time on each link is consistent with the load on that link.

We opt instead for simple constants to represent delay time on each link.

Updating new priorities on the basis of information delivered to users is an order of magnitude more challenging. We do not develop information about intelligence products other than the Red order of battle. Serious updating of information needs would require a broader inquiry into higher-level inferences about Red intent. Analytic tools are not available to model the development of higher-level inferences. Effectively placing their development in the context of statistical decision theory—asking what and how much information to collect before making a decision—is too demanding in the absence of a basic analytic framework for constructing such inferences.

Similarly, predicting the effects of the quality of information simulated by our model on future combat outcomes is still more difficult. Such an effort would require analysis of higher-level inferences and their combat value. We have not found adequate analytic tools to model either.[4]

The alternative approach we have adopted is consistent with our desire to measure the effects of incremental changes in an intelligence system. Suppose we are interested in the incremental value of a new collector. We first represent the intelligence system as it would work with that collector. We simulate an engagement and determine what Red units do on the deep battlefield. As part of this, we set a ccllection schedule and set of priorities for unit-attributes consistent with the information a Blue commander would want during this engagement. We determine the delay times that would occur on communication links and in processing activities over the course of the engagement. We then fix the time series for all of these factors through the course of the engagement:

- behavior of Red units
- collection schedule
- priorities on unit-attributes
- delay times in processing and communication.

This forms a baseline. We use it to simulate information flows through the intelligence system and measure how they affect the quality of information that the system yields. *Given* these times series, we

[4] Work is underway at RAND to model such feedbacks. It uses an accounting framework that accommodates fairly simple subjective judgments about how intelligence development affects the quality of higher-level inferences and assumptions about how that affects combat outcomes. Although this approach can be useful in answering some questions, we do not expect it to help us address our main concern: the effect of incremental changes in the Army's intelligence system on its ability to pursue deep battle.

now remove the new collector and simulate information flows again. We compare the resulting measure of information quality with that for the baseline and use it as a basis for judging the incremental value of the new collector. Similar adjustments relative to this baseline could be used to judge the incremental value of new processors and communication lines.[5]

When a new system has a small effect on the scenario as a whole, such an approach yields a good approximation of the measure we would get if we modeled all feedbacks in the system completely (and properly).[6] As effects become larger, the approach becomes suspect. It should not be used to compare the performance of fundamentally different intelligence systems.

We recognize, then, the importance of a wide range of feedback loops in the development of intelligence. But we do not have analytic tools to help us to predict how changes in an intelligence system would affect each of these feedbacks. Whatever baseline case we use must adequately reflect the effects of such feedback. We have chosen an analytic approach that allows us to capture all effects of such feedbacks in the baseline itself.

If we cannot model these feedback loops, how can we determine what feedback should occur in the baseline? The simple answer is that we rely on military judgment about some feedbacks and use sensitivity analysis to examine the importance of others. VIC embodies military judgment about the feedbacks of information quality on collection

[5] The approach is analogous to the use of a Paasche index to measure price, quality, and welfare changes in economics. For example, to measure the aggregate change in price level in the economy, a Paasche index would hold constant quantities following the change and use them to weight price changes. Our approach similarly holds constant a wide range of variables at their values after a change and uses them to judge the effect of a selected change in the intelligence system on the quality of information it produces. An approach analogous to a Laspeyres index, which holds constant circumstances before a change to judge the effects of that change, could be equally appropriate for our analysis. Where we expect a change to have large and widespread effects, Laspeyres and Paasche analogs could be used to bound the size of the effect of information quality that interests us. Although our approach could allow such an approach, the construction of a baseline based on VIC is cumbersome and demanding; we concentrate on the Paasche analog for now.

[6] In essence, we are using analogs to the first-order terms of a Taylor series expansion around the baseline case to approximate the effects of a departure from the baseline case. If M were a figure of merit, $M = f(x_1, \ldots, x_n)$, where x_i are measures of inputs relevant to collectors, processors, and communication links, and f ( ) were suitably well-behaved,

$$\Delta M = \Sigma f_i \Delta x_i + \epsilon ,$$

where ϵ is a sum of higher-order terms that are close to zero, unless changes in x_i significantly affect f_i *and* Δx_i are large. Without claiming that we are using a well-behaved function f ( ) that translates policy-relevant changes in an intelligence system into changes in a figure of merit, we can say that our approach uses similar logic.

schedules and information quality on combat outcomes. Each VIC scenario is carefully crafted, using extensive human intervention, to ensure that it is consistent with prevailing military judgment.[7] A similar effort could be used to set priorities for unit-attributes that are consistent with a scenario. Military judgment does not offer a consensus on delay times in processing and communication.[8] In all likelihood, a range of delay times should be simulated as part of any analysis. The range of basic uncertainty probably exceeds the change in these delay times that might result from an incremental addition to an intelligence system. Uncertainty about delay times almost certainly exceeds the size of any feedback effects relevant to delay times.

In sum, VIC provides a great deal of information about the baseline. We could rely on alternative combat simulations if that were appropriate. If we can assume that VIC handles feedback in a way that satisfies military judgment, this approach allows us to avoid modeling feedback effects. However, this approach is not appropriate to judge the effects of changes in the intelligence system that could influence the underlying scenario.

AN INTEGRATED VIEW OF INFORMATION FLOWS

Figure 3 presents a network diagram of the simplified system. It includes four collection sources, six processors, 12 communication links (labeled, "Ci"), and two users. Table 1 explains the collectors, processors, and users. For our purposes, GRCS will generate three data streams, and we model them as though they were from separate collectors. JSTARS, which the Air Force will actually operate, is expected to allow full Army participation in its use and to be well integrated into corps intelligence development. Systems over which the Army has no direct control could be represented just as easily. The processors and their connectivity in the system are notional; exact, planned arrangements in the U.S. corps could be represented in a network model without any more complexity than that shown here. ASAS/ENSCE would presumably be composed of a similar set of

[7] That is not to say that VIC's treatment of these feedbacks could not be improved. The TRADOC Analysis Command (TRAC) at White Sands Missile Range, New Mexico, is pursuing a program of research to develop improved intelligence models that can be applied within the context of VIC. As such improvements occur, we expect to be able to use improved output from VIC that better reflects feedbacks of this kind.

[8] Empirical data on current delays in European command post exercises and planning factors for improvements planned for the mid-1990s range over two orders of magnitude or more. Based on our interviews in Europe, informed judgments on likely delay times in the mid-1990s range over more than an order of magnitude.

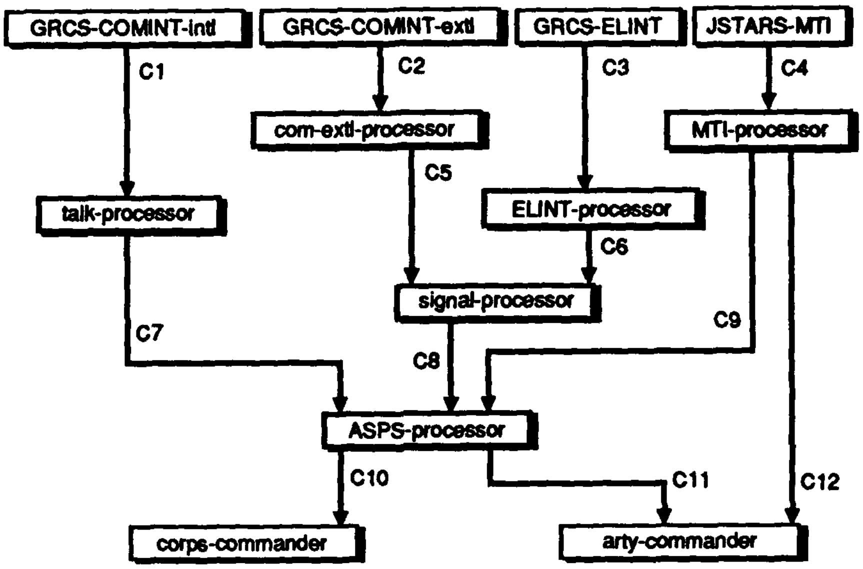


Fig. 3—Simplified corps intelligence system

processors and the software they use to communicate and manage each others' data. Links to users reflect the delay times before intelligence products reach the people who need these products to support the corps commander or use the ATACMS. Communications link C12 contains a quick-fire channel that moves data from JSTARS to the corps' artillery system fire control information system and then uses that system to move data to a prearranged ATACMS launcher on hot stand-by.

The elements of the network diagram in Fig. 3 look quite similar to the nodes and arcs associated with Steps B through F in Figs. 1 and 2; Fig. 3 does not capture Step A, which moves information on priorities from users to collectors and processors. Because we do not explicitly model the feedback processes in intelligence development, we do not treat such an information flow as part of the network model that we develop.

In our model, the intelligence system shown in Fig. 3 moves information from the top nodes to the bottom nodes in the context of an external environment that has four basic elements:

- The behavior of Red units on the deep battlefield, or Red "ground truth." VIC generates this information in the course of

Table 1

ELEMENTS OF A SIMPLIFIED CORPS INTELLIGENCE SYSTEM

Element Name	Description
Collectors	
GRCS-COMINT-intl	GRCS COMINT internals. GRCS is a future Army aerial platform that will carry COMINT and ELINT collectors. This is the data feed on the internal content of COMINT.
GRCS-COMINT-extl	Data feed on the external signatures of COMINT from the GRCS. It comes from the same sightings used for the data feed above.
GRCS-ELINT	ELINT data from GRCS. It comes from the same orbits used to generate the data feeds above, but not necessarily the same individual sightings.
JSTARS-MTI	JSTARS Moving Target Indicator, a future joint Army-Air Force form of radar IMINT collector.
Processors	
talk-processor	Ground station with interpreters of the content of COMINT internals.
com-extl-processor	Automated correlator for COMINT external signatures.
ELINT-processor	Automated correlator of ELINT data.
MTI-processor	Ground Support Module (GSM) for the JSTARS MTI.
signal-processor	Automated correlator for Signals Intelligence (SIGINT).
ASPS-processor	All-Source Production Section (ASPS) of the Corps Tactical Operations Center (CTOC).
Users	
corps-commander	G-2 (intelligence officer) on the corps commander's staff.
arty-commander	ATACMS fire unit operator, via the gateway to the artillery commander's fire control information system.

simulating the combat scenario we use to establish our baseline. We need run the VIC scenario only once. We place information from this run in a file that our simulation can use repeatedly without running VIC again.[9]

[9] When appropriate, we can supplement the ground truth included in a VIC baseline in a way that facilitates study of the effects of deception. We have designed this capability into our modeling approach but have not attempted to implement it. The appendix provides some details.

- The Blue collection schedule. VIC provides a baseline collection schedule and utilities that allow the alteration of this schedule. Alterations can then affect the baseline simulation if the VIC baseline is unsuitable. Whatever schedule we choose serves as an input to the baseline VIC sce.ıario. Running that scenario generates a file of information on what portion of Red ground truth the Blue collection plan detects. Our simulation can use this file repeatedly without running VIC again.
- The Blue commander's priorities for unit-attributes. For our model, an order-of-battle specialist specifies these based on what he observes in the VIC baseline scenario. The model places these in an exogenous file that our simulation can access at any time to establish priorities. Priorities can change over the course of a simulated engagement.
- Delays associated with Blue processing and communication. We place information on delay times, by priority level, for each processor and communication link, in ιn exogenous file that our simulation can access at any time. Delay times can change over the course of a simulated engagement.

In sum, the information reflected in these four elements is not affected by information flows that we simulate in the network described above. The information reflected in these four elements, taken together, effectively forms an exogenous environment in which we can simulate information flows through the network.

We transform the contents of the file that VIC generates to show what battlefield information the collection plan has gathered into a list. The list contains an item for each event and time when a collector receives new battlefield information on a particular unit-attribute. The list orders these items by the time of receipt, starting with the earliest. Our simulation works its way through the list. Each item initiates a transaction for the relevant collector in our network model. That transaction initiates a series of transactions that push this new information through th϶ network. Transactions that depend on information about unit-attribute priorities or delays use information from the relevant exogenous files to determine their values. Ultimately, the new information generates transactions that send new conclusions to the final users in the model. The simulation records information about these conclusions and the time when they reach each user. It generates a time series of information about conclusions on each unit-attribute. We can then analyze these time series to measure the quality of intelligence associated with that unit-attribute over the course of the simulated engagement.

We can indicate how we look at information flows with and without a particular element of the intelligence system. Suppose we are interested in how external COMINT data from the GRCS affect information flows in an intelligence system. We would analyze this in the following way. First, create the exogenous environment of Red behavior, collection, unit-attribute priorities, and delay times. Second, set up the network shown in Fig. 3 and generate information flows through it using the external environment. Third, set up another network that excludes the GRCS-COMINT-extl collector. Generate information flows through this variation on the original network using the same external environment. Use the difference in information flows as the first step in judging the contribution of the GRCS-COMINT-extl collector.

This example illustrates a fundamental aspect of our model. In a particular analysis, the external environment is fixed once and for all. It defines an analytic baseline. To change an intelligence system, we adjust only the network of collectors, processors, and users. We start with a network that includes all parts of the system we wish to study and then isolate the incremental contribution of any element in the system by eliminating it from the network. This approach provides a simple way to use what can become a rather complex analytic tool.

SUMMARY

This section presents our approach to modeling information flows through an intelligence system. To model any intelligence system, *we focus on a fairly simple flow of information*: information about Red unit-attributes. That is, any flow of information concerns a unit's name, type, echelon, location, speed, direction, effectiveness, or activity. Intelligence managers first send information on priorities to collectors and processors from users in the form of a collection schedule and priorities for the collection and processing of information on specific unit-attributes. Second, collectors gather information on Red battlefield activity and send it to processors through a communication system that is responsive to priorities. Third, a series of processors within the system move information from different places to a single processor that develops final products, which it sends to users through a communication system that is responsive to priorities. Finally, users receive intelligence products. They (implicitly) then determine new information priorities.

Our approach reflects a simplified version of the basic cycle implied by the four steps above. *It measures the effects of changes in an* *intelligence system relative to the performance of that system in a baseline.* The baseline is based on a specific combat scenario, developed by the Army, and reflects a series of feedback loops that we do not model explicitly—the effect of the quality of intelligence on combat outcomes, the effects of communication and processing loads on delays, and the way information affects users' information priorities. Rather than model these feedback loops explicitly, we accept the loops as they exist in the baseline simulation and consider only changes in an intelligence system that will not change these feedbacks in a major way.

Given this approach, *we model information flows from priorities to collection to processing to users as one-way flows through a network.* The network represents collectors, processors, and users as nodes and communication links as arcs. We change an intelligence system by changing elements of this network. Information flows through this network in the context of an exogenous environment that defines the analytic baseline. It defines the behavior of Red units on the battlefield. It defines Blue's collection schedule, initiating information flows in the network by supplying information on observed unit-attributes to collector nodes. It defines unit-attribute priorities for collection and processing that govern how these elements of the intelligence system treat information. And it defines delays in processing and communication elements as a function of priority. The behavior of Red units, the Blue collection schedule, unit-attribute priorities, and delay times reflect a wide variety of feedbacks that we accept as being modeled properly in the baseline.

As information moves through this network under any particular regime of priorities and delays, the intelligence system changes its quality. Our ultimate interest is in the quality of information reflected in intelligence products that the system provides to final users.

IV. MODELING INFORMATION QUALITY IN AN INTELLIGENCE SYSTEM

An intelligence system receives new information about Red units and uses this information to build increasingly complete intelligence products that present inferences about the status and behavior of Red units on the deep battlefield. In this process, we can think of information as flowing into the system through collectors and becoming embedded in the intelligence products that the system generates. At each point in the system, how much does the new information about a Red unit-attribute, embedded in the interim intelligence product that a Blue processor or user receives, contribute to the quality of intelligence that that processor or user maintains on this unit-attribute?

DEFINING AND MEASURING THE QUALITY OF INFORMATION

Our analysis revolves around the ability of an intelligence system to define the Red order of battle during a campaign. In particular, we are interested in the system's ability to confirm the presence of Red units on the deep battlefield and determine the values of the important attributes associated with these units.

Consider the value of a particular attribute of one Red unit, a "unit-attribute value." The record of this value over time effectively defines ground truth for this unit-attribute. Elements of the Blue intelligence system—collectors, processors, and users—maintain subjective probability distributions that define their "beliefs" about this value over time.

They generally do not do this consciously. For example, a good order-of-battle analyst, when asked about the identity of a unit, will not answer, "We believe there is a 45 percent chance that it is the 5th Guards and a 55 percent chance that it is the 17th Guards, sir." But he will often say, "It's either the 5th or the 17th Guards, sir. We're getting information to clarify that now. We don't want to hazard a guess now, but if you pressed us, we hold a slight preference for the 17th." And he would explain why. Order-of-battle analysts do use ellipses to express their beliefs about the location of certain Red units. If pressed, they will agree that, say, an 80 percent chance exists that actual location lies within such an ellipse. Other analysts will agree that they do not post a unit-attribute value in their databases until they are at least about 80 percent sure that they have the right value. In sum, although formal databases generally do not include explicit information on subjective probability distributions about unit-attributes, good order-of-battle analysts are acutely aware of the uncertainties they perceive with regard to Red unit-attributes and the importance of that uncertainty to battle planning and collection management.

We use an approach based on formal subjective probability distributions to capture the implications of this more heuristic understanding of uncertainty for decisionmaking. With this in mind, the best way to think about how accurate beliefs are in an intelligence system depends on whether the unit-attribute in question takes categorical or continuous values.[1]

Consider a unit-attribute with categorical values first. This holds for all the attributes considered in Secs. II and III except location and speed. How much weight does a Blue element's subjective probability distribution for this unit-attribute assign to the value that actually defines ground truth over time? That is, it is natural to suggest that a belief is more accurate, the more weight it gives to the value of an actual unit-attribute. With this in mind, we propose the subjective probability that a Blue element assigns to the value of an actual unit-attribute during a particular period as a measure of effectiveness for the intelligence product that that element maintains on this unit-attribute. For example, if the name of the unit that order-of-battle analysts above are discussing is actually the 5th Guards, the quality of information associated with their beliefs during that discussion is 0.45. Aggregating across unit-attributes and time can yield aggregate measures of effectiveness for relevant elements in the intelligence system.

The attributes of location and speed take continuous values. In these cases, we redefine the attributes to have categorical values by breaking their continuous values down into discrete ranges. For example, we ask, "Can Blue target forward elements of the unit with a dumb weapon—that is, does Blue know location to within 100 meters?" The measure of effectiveness is the probability assigned to a circle with a radius of 100 meters around the actual "relevant" location. Alternatively, we ask, "Can Blue target elements of the unit with a smart weapon—that is, does Blue know location to within a kilometer?" An analogous measure of

[1] A unit-attribute takes categorical values if the subjective probability distribution that defines beliefs about it can be expressed in terms of categories—if it is discrete. For example, a unit is either an armored unit or a motorized rifle unit or some other discrete kind of unit. A unit-attribute takes continuous values if the subjective probability distribution that defines beliefs about it is continuous. For example, the center of mass or forward elements of a unit can lie anywhere in space; we do not typically measure their location as being in Location A or Location B.

effectiveness emerges. Or we ask, "Does Blue know the general location of the unit's center of mass—that is, can Blue place it within 10 km?" Taken together, these three questions could generate a nested set of measures that assign subjective weight to circles with radii of 100 m, 1 km, and 10 km. For simplicity, we use only one circle to define the quality of information associated with location. We apply a similar approach to speed.[2]

The reason for treating attributes with categorical and continuous values differently is really that the typical number of categories for categorical variables is low. For example, direction can be "north," "east," "west," or "south." Hence, we are less concerned about the proximity of differing values for attributes with categorical values than for those with continuous values. If the proximity of values presents a problem with categorical values, we can look at the subjective probability assigned to any combination of values for an attribute to examine the issue of proximity. For now, we will consider only probability associated with the actual value of such an attribute.

This *definition* of quality is only one of several that analysts might consider. For example, an alternative definition might emphasize the importance of "regret." We could ask how much subjective probability Blue places on the value of a unit-attribute that would lead to the worst outcome for Blue planning, given Red's true behavior. The measure we use is related to such a measure; the more weight Blue places on a true value, the less it can place on a value that would endanger its planning. Nonetheless, to determine what values are most damaging to Blue at any point in space and time on the battlefield, a measure based on regret would require a sophisticated understanding of the context in which Blue observes Red unit-attributes. Our approach does not require such information.[3]

[2] Other approaches are possible. For example, we considered an option that identifies the parameters of the subjective probability distribution for location or speed and asks how these change as new information accrues. The approach uses formal statistical methods to update information about these parameters as empirical data accumulate. The approach we use is simpler and allows us to use available information on the quality of collection as well as this alternative would. As better information becomes available, however, a more formal approach may be warranted. For details, see Bunn, 1984, pp. 127–141; cf. Hogg and Craig, 1970, pp. 111–114.

[3] In the context of decision theory, we are making a strong assumption about the loss *function associated* with information about unit-attribute values. That function takes one value for correct unit-attribute values (or, for unit-attributes with continuous values near the correct value) and another value for all other unit-attribute values. This implicit assumption makes our simplified approach to defining the quality of information possible. Given the level of detail modeled in our approach, such simplicity is especially appropriate. While the loss function is somewhat arbitrary, then, no alternative is compelling enough for us to sacrifice the simplicity it offers. For a useful discussion of this issue, see Zellner, 1987, esp. pp. 291–298.

Some readers have commented that they find "quality" inherently difficult to imagine quantifying. What we call "good" or "high quality," they think of more as sharp resolution, a characteristic that may or may not be the same as a command staff's general assessment of quality in a particular situation. The measure we use offers a simple and useful view of a richer perception of quality. Given the level of detail that we model, this simplicity is appropriate. Nonetheless, as we discuss levels and determinants of "quality," we use a very specific definition of the quality of information.

Our definition of quality raises some specific potential problems that we can address in the context of our approach. For example, how can we use this approach to judge an intelligence system's ability to avoid seeing things on the battlefield that are not there? Radars and radios can easily generate false images, even when Red does not intend them to. Through *maskirovka*, Red can also attempt to persuade Blue that certain units are present when they are not. At this point, our analysis does not address these issues. The appendix discusses ways to introduce them by adding false elements to ground truth. The measure of effectiveness we would use here is the subjective probability Blue components assign to the hypothesis that these observations are in fact false. That is, within the context of our model, we should be able to judge Blue's ability to recognize these false images in exactly the same way we judge Blue's ability to see real units on the battlefield.

Another problem is that values assigned to different attributes of a unit or to attributes to related units can depend on one another. For example, if Blue believes the speed of a unit is "zero," Blue should also believe that its direction is "stationary" and that it cannot be engaged in such activities as "column march."[4] Similarly, if Blue sees two similar units on a major road and believes one is moving at a particular speed, Blue will probably believe that the other unit is moving at a similar speed. The approach we propose can allow such dependence, but its presence will not be immediately evident even if we include it because we do not model complete subjective probability distributions and the weights given to all values in these distributions. We consider only the probability assigned to the actual value of each attribute separately. To the extent that such a probability depends on probabilities assigned to more than one attribute in a unit or to a unit-attribute across units, that will have to be reflected in rules used to update these probabilities. As currently formulated, our approach does not incorporate such rules.

[4] In fact, "stationary" is not an allowable value for direction in the model. We use only the four cardinal directions to define direction.

INTELLIGENCE FUSION IN A COMPLEX SYSTEM

To establish the values of each of the unit-attributes of interest, an intelligence system must bring to bear information from several sources and "fuse" that information into a subjective probability distribution. To do this, elements within a Blue intelligence system continually pose hypotheses about Red behavior and use empirically based information to test them. Elements of Blue intelligence maintain complex but typically implicit models of Red unit behavior that they use to frame hypotheses about the values of its attributes.[5] Blue expects each unit to have a fairly detailed list of equipment. And given this equipment, the terrain in which the unit is operating, and what Blue thinks the Red commander's plan is, Blue expects Red to move and operate that equipment in a fairly predictable way.

Blue's beliefs about Red can be framed as a set of joint, testable hypotheses. For example, the All-Source element of the Blue intelligence system may expect a Red unit to move from a bivouac area in a particular pattern down a road, then deploy off-road in separate columns and deploy its air defense and artillery units in particular ways in the terrain it occupies. The Blue All-Source element can then use HUMINT to monitor the unit's activity and identity as it passes certain check points, use MTI to watch the unit's movement and perhaps say something about the kinds of vehicles it has, use ELINT to watch its air defense deploy, and use COMINT to monitor radio traffic, looking for the kinds of radios Blue expects this unit to have and listening for clues about the unit's identity and mission.

We can think of a Blue element's view of this Red unit as a complex if-then statement. Blue intelligence thinks, "If my HUMINT says X_1, MTI says X_2, ELINT says X_3, and COMINT says X_4, then this Red unit is Y_1, and it is executing a maneuver, Y_2, that we would expect if the Red commander's plan were Y_3." In fact, pedagogical discussions of fusion often explain it in this form: If you observe $[X_i]$, then you can conclude $[Y_i]$. This description presents three problems for someone attempting to model changes in the quality of Blue intelligence associated with fusion.

[5] As is typically true of experts, order-of-battle analysts are conscious of how their expertise is organized in only the roughest sense. They use highly complex models, which they continually update, to make sense of the masses of data they must assimilate. But we can only begin to fathom the structure and nuances of these models through detailed questioning about why they make particular decisions. As often as not, they can explain why they make a particular decision, but they do not consciously execute the logic they use to explain their decisions when they make them. In sum, the expertise of order-of-battle analysts is no easier to observe and model than the expertise of the practitioners of other complex arts.

First, it presumes that the if-then rule is correct and a Blue element understands Red well enough to specify precisely how it organizes each unit and how it uses each unit in any piece of terrain under any plan. This obviously asks too much. The outcomes may be "likely;" if so, it makes sense to ask how much uncertainty remains about conclusions when all the antecedent conditions are met.

Second, it presumes that a Blue element has all the information specified by these antecedents with a high degree of confidence. Blue doctrine recognizes that such confidence is not typical and offers a formal way to specify the quality of information sources.[6] It does not indicate what to do when information is poor. In practice, experienced order-of-battle analysts become more conservative about their conclusions when they doubt the quality of their information sources.

Third, a Blue element—even the All-Source activity—rarely has all of the information called for by antecedents, at any level of quality, and almost never gets all the information at the same time. Order-of-battle analysts in each element of the Blue system must reach conclusions on the basis of limited information and update these conclusions as new information arrives. Unfortunately, by that time, older information is dated and of less value. At any time, good order-of-battle analysts recognize that uncertainty about conclusions is unavoidable.

Given Blue's hypotheses about a Red unit, then, Blue has more confidence in its hypotheses, the closer they are to prior Blue beliefs about how the Red unit would behave, the more empirical information Blue receives that is consistent with these hypotheses, and the higher Blue's confidence in the information it receives. Over time, Blue poses hypotheses, tests them, and alters them to reflect Blue's confidence in its conclusions on the basis of the information it has. Each Blue element uses its limited resources to fuse information in a way that maintains as high a level of confidence as possible in important parts of its Red order of battle.

We cannot attempt to model this behavior in terms of an explicit set of probabilistic if-then rules for each Red unit and contingency that each Blue element might anticipate. The complexity of such rules and of the relationships among the elements of the Blue intelligence system help explain why tactical fusion is at least as much an art as a science, and human order-of-battle analysts dominate automated systems in all but the simplest fusion tasks. Our interest is in the quality of the final product of a complex process of repeatedly forming and testing hypotheses; a general understanding of that process can help us posit a

[6] Order-of-battle analysts do not appear to use this doctrinal method during exercises, but they are quite aware of what quality they can expect from each of their sources. For the doctrinal approach, see FM 34-1.

simple set of rules with which to simulate changes in the quality of the final product. This discussion suggests four simple rules: The quality of the final product of this process increases as Red units behave more predictably; increases each time new, relevant, good information arrives; increases as the quality of new information increases; and decreases as time passes without new empirical information. A corollary of these four rules, taken together, is that the quality of the final product of this process can decrease when new information is of low quality or deceptive.

Taken together, these rules have implications for intelligence fusion and its effect on information quality that are consistent with those emerging from viewing intelligence fusion in terms of the following information flow in a set of databases. Each database contains the subjective probability distribution that an element—collector, processor, or user—in an intelligence system maintains for each unit-attribute. If this database receives no new information, its probability distributions gradually become more and more diffuse over time. Any time it receives a new piece of information, it (implicitly) invokes a probabilistic if-then fusion rule like that posited above. The rule recognizes the quality of information residing in the database when new information arrives and the quality of the information arriving. This quality and the time it takes an element to transform inputs into outputs affect the level of confidence it places on its outputs. This element passes along information about this level of confidence when it sends its output to other elements in the intelligence system. The quality of information that any element receives reflects the quality of that information when it was generated and how much time has passed since it was generated.

This logic lies at the heart of our simulation of how information quality changes in an intelligence system. We focus on the one aspect of this system that interests us—the quality of information. We measure this for each Red unit-attribute as the subjective probability that an element of the system associates with the correct value of that unit attribute. We think of collectors and processors in the intelligence system as production activities that transform the quality of information they receive into a level of quality for information they produce. That is, without addressing any of the specific probabilistic if-then rules that reside in this system or the actual inferences they generate, we focus on a particular definition of information quality. The availability of a simple way to degrade the quality of information over time and a formal technique for updating databases that can transform the quality of inputs into a measure of the quality of outputs makes this possible.

DEGRADING THE QUALITY OF INFORMATION OVER TIME

The simple logic above suggests that the quality of information in an intelligence system should fall as time passes without access to new empirical information. To maintain good information about a Red unit over time, Blue must be able to (1) associate new data with the right Red unit; and (2) given (1), apply these data to its implicit models of the Red unit to test its beliefs about that Red unit's behavior.[7] Both of these tasks become more difficult as time passes without new empirical data on the Red unit.

Falling Ability to Infer Red Behavior

Consider first a Blue element's ability to make statements about Red behavior in the absence of new information. The following definitions will facilitate our discussion. Let A_t be the actual (categorical) value of a unit-attribute at time t and $p(A_t)$ the subjective probability a Blue element assigns to this value at time t. Suppose the value of A_t changes over time. The Blue element can use its (often implicit) models of a unit-attribute to predict how this change will occur. But as time passes, the Blue element's subjective probability distribution becomes more and more diffuse, causing $p(A_t)$ to fall.[8] The better Blue's model of the unit-attribute in question, the more slowly it will fall. For example, the values of unit names, types, and echelons are fairly easy to model because they rarely change; unit speeds and locations present a more difficult challenge because they can change repeatedly. Similarly, major command posts may be stable for days at a time in location, effectiveness, and activity; surface-to-surface-missile (SSM) launchers may be moving and changing activity from minute to minute. The point is that information decays at different rates for different unit types and unit-attributes. We allow for such differences in our simulation. As information decays, the probability that Blue maintains accurate information about Red falls.

[7] For simplicity, we phrase most of this argument in terms of unit-attributes with categorical values. The argument applies equally for unit-attributes with continuous values if we replace "the correct or actual value of a unit-attribute" with "values that lie within some specified distance of the actual location or speed." For example, we could use any location within 100 meters of the true location or any speed within 1 km/h of the true speed.

[8] In cases where Blue analysts have assigned most of their subjective probability to the wrong value of an attribute, an increasingly diffuse probability distribution could actually increase the quality of information as we define it. We expect this to happen in an intelligence system, but we do not expect it to be a typical or dominant occurrence.

Unit Association

Over a period of time, Blue receives information about any particular unit from several collectors or from several orbits of the same collector. To bring all of this information to bear, a Blue element must associate new data with the right unit. The better its ability to do this properly, the better will be the Blue element's knowledge about units for which it is acquiring information *and* the lower will be the probability that the Blue element improperly associates these new data with other similar units nearby, thereby polluting its inferences about these other units.

A Blue element's ability to associate new data with the right unit depends on two factors:[9] (1) the time that has passed since the element last received data on the area where the unit was located, and (2) the proximity and similarity of units that could be confused with the unit. We treat (1) as simply as possible. When we examine factor (1), two effects concern us. One is failing to apply new data on a unit to a Blue element's knowledge about it. The other is applying new data on a unit to the element's knowledge about another unit. Whenever new data become available on a unit, we should expect both factors to pose a problem for the element's knowledge about that unit. When an element receives new data on one unit that it might misassociate with similar nearby units, it will probably receive new data on these other units as well. Hence, we can assume that delay times between observations are similar for all of these units and we can treat both types of effects on a Blue element's knowledge about any particular unit from the point of view of that unit.

The rate at which these difficulties cause the quality of information to degrade presumably differs across units. The rate is also probably related to rates for individual attributes discussed above, but these are likely to differ within any unit. For example, Blue is highly likely to lose the location of a fleeting target such as an SSM launcher but continue to know that the unit exists and is active on the battlefield. Hence, loss of a unit's location should not mean that the unit itself is lost. However, Blue confidence about a Red unit's name can persist long after all traces of the Red unit have disappeared from the battlefield. How to relate these rates of decay requires careful attention.

Exponential Degradation

The processes that lead the quality of information to decay over time—the fall over time in a Blue element's ability to infer Red

[9] For a useful discussion of this complex problem, see Blackman, 1986.

behavior or to associate data with the right units—are obviously complex. We cannot hope to model them directly. We prefer instead a simple approach that uses a single parameter to indicate how quickly quality falls over time in particular circumstances; we use exponential decay to represent the loss of information that occurs in the absence of new information or data association:[10]

$$\frac{1 - p(A_{t_1})}{p(A_{t_1})} = \frac{1 - p(A_{t_0})}{p(A_{t_0})} \exp[D(t_1 - t_0)] \qquad (4.1)$$

where t_0 and t_1 are two points in time, no new information arrives and no data association occurs during the period from t_0 to t_1 , and D is an exponential decay rate (defined positive) specific to a unit-attribute.[11] Using this functional form facilitates calculations we discuss in a moment. The odds ratios shown ᵗake values from zero (for $p(A_t) = 1$) to infinity (for $p(A_t) = 0$). Note that if a Blue element is ever totally wrong about a unit-attribute and $p(A_{t_0})$ equals zero, the relationship in Eq. (4.1) is undefined. For practical purposes, we will hold each Blue element's level of confidence above this level; but we can allow $p(A_t)$ to approach zero closely without posing a problem. Note also that this form of quality degradation can never drive $p(A_t)$ to zero, which has important implications below.

Although Eq. (4.1) defines decay in information quality for only a discrete period from t_0 to t_1, any scenario comprises a series of such discrete periods and the type of decay represented in Eq. (4.1) occurs continuously for each Red unit-attribute through the course of any scenario. It is relieved only at such time, like t_0 and t_1, when new information arrives at a Blue element and new data association occurs.

[10] Our decision to use exponential decay is based on similar logic used in analogous decisions to represent the depreciation of capital assets or the growth of technology exponentially. Many methods exist to model the depreciation of individual capital assets; similarly, individual technological innovations can have many different effects on costs and capabilities. But when analysts consider the depreciation of many assets together over time, exponential depreciation serves remarkably well to represent this depreciation. Similarly, when analysts consider the effects of many innovations together over time, exponential growth works well to represent their cumulative effects. In our situation, where data are scarce and we are interested more in the effects of many information processing actions taken together than in the effects of any one action, the exponential model offers a simple approach that is likely to represent behavior well in an intelligence system.

[11] The decay rate could also reflect values of unit-attributes, the time during a campaign, and the unit's location on the battlefield. We have not attempted to model such subtlety at this time, but it could be modeled fairly easily in the future.

A BAYESIAN APPROACH TO UPDATING

We now must consider what happens when new information arrives at a Blue element to (potentially) offset the general degradation of information over time. Consider the following situation. The database in an element of the Blue intelligence system contains a subjective probability distribution for a particular unit-attribute at a certain time. New information about that unit attribute arrives. How would this element use the new information to update its database?

Bayes' Theorem is a simple but powerful statement that provides an internally consistent way to order and update subjective probability. If we state the quality of new information properly, it allows us to calculate how that new information, when added to a database, affects the quality of information in it. This does not imply that the elements of an intelligence system use Bayesian techniques to change their perception of uncertainty when they accept new information. They may or may not.[12] We are not modeling their perception of uncertainty so much as we are modeling their ability to use new information to update their beliefs and the effect that that new information has on the quality of their beliefs.

In a Bayesian context, we can state our situation as follows. At a certain time t, $p(A_t)$ prevails as the subjective probability that a Blue element assigns the correct value of a unit-attribute. New data, x, arrive.[13] We now want to know the updated probability the Blue element assigns to the event that A_t holds, given the availability of x, $p(A_t \mid x)$. Bayes' Theorem states that, if the Blue element accepts x

[12] Selected automated parts of the Blue intelligence system do use formal Bayesian updating methods and those that do not, use methods that approximate Bayesian methods. In their own activities, order-of-battle analysts do not consciously use Bayesian techniques. Experienced analysts may view the effects of updating on quality in a way that is consistent with Bayesian techniques, simply because experience has led them to do their jobs well. Actual observation of order-of-battle analysts in peacetime exercises reveals that their experience varies widely. With enough exercises, less experienced analysts will presumably achieve the sophistication of their more experienced colleagues; but if war started today, they would not do so in the crucial opening days of combat. Whether the Blue intelligence system behaves the way Bayesian methods would dictate or not, we can use Bayesian methods to measure its performance.

[13] x can consist of any kinds of data. For our purposes, the quality we associate with these data applies to the moment when a Blue element processes them. If they have just arrived from the battlefield, they contain all the information they could contain. If they have been delayed or stored in the Blue element's database before this moment of processing, their information quality must reflect the fact that time has passed since collectors gathered them from the battlefield. With this in mind, x can include information of differing vintage, so long as the quality of each vintage is properly degraded.

and combines it with information it had previously, the follow'ng holds:[14]

$$p(A_t \mid x) = p(A_t)p(x \mid A_t)/[p(A_t)p(x \mid A_t) + p(O_t)p(x \mid O_t)] . \quad (4.2)$$

$p(x \mid A_t)$ is the probability that the element would receive the new data, x, if A_t were true. $p(O_t)$ is the probability that A_t is not true, and $p(x \mid O_t)$ is the probability that the Blue element would receive the new data, x, if A_t were not true. Equation (4.2) provides a direct method for updating our measure of how well a Blue element chooses the value of a unit-attribute when it has categorical values. A more revealing way to present this expression is:

$$\frac{1 - p(A_t \mid x)}{p(A_t \mid x)} = \left(\frac{1 - p(A_t)}{p(A_t)} \right) \left(\frac{p(x \mid O_t)}{p(x \mid A_t)} \right) . \quad (4.3)$$

The first ratio on the right is the odds ratio that prevails before new data arrive; it is the "a priori" odds ratio. It reflects whatever degradation has occurred in the Blue element's database because of the passage of time up to the moment when new data arrive. As noted above, it can take values from zero to infinity. The ratio on the left is the Blue element's updated odds ratio based on the new information; it is the "a posteriori" odds ratio. It can be interpreted in a similar way.

We use the second ratio on the right to transform an a priori into an a posteriori odds ratio. That is, *if* a Blue element accepts a new datum and uses it to update its database, this ratio shows how this new datum affects the quality of the database. The ratio may look familiar as the likelihood ratio one would use to compare two hypotheses with the data x.[15] As suggested above, the Blue element in fact uses these data, together with its a priori beliefs, to compare two hypotheses: that A_t is true, and that it is not. The likelihood ratio can take any value from zero to infinity.

Acceptance of new empirical information improves or degrades the performance of the Blue element depending on which of the probabilities in the likelihood ratio is larger. If x can occur only if the unit-attribute takes the value A_t—that is, if $p(x \mid O_t) = 0$—this ratio goes to zero and the updating process immediately drives the odds ratio on the left to zero. The new information perfectly discriminates between the value that the unit-attribute actually takes and all other values. If x is far more likely to occur for values of the unit-attribute other than the

[14] For a simple exposition, see Raiffa, 1968.

[15] Cf. Zellner, 1987, pp. 291-298.

one actually occurring than it is for its actual value, the ratio can be large, reducing the weight that Blue places on the proper value of the unit-attribute. Because this likelihood ratio measures a Blue element's ability to use new data to discriminate between two hypotheses, we will refer to it as a "discrimination ratio."

Why would a Blue element accept new information that degraded the quality of information in its database? Ideally, Blue would never accept information whose discrimination ratio is larger than unity. But Blue cannot measure this discrimination ratio directly. On the basis of their capabilities, Blue analysts will decide what information to accept. Suppose Blue analysts hypothesize that an observed Red unit is an artillery unit when in fact it is an armored unit. These analysts may tend to collect and accept information that is consistent with their hypothesis. Such information will have discrimination ratios higher than one. When analysts make this mistake, they progressively accept data that confirm their expectations, driving $p(A_t)$ down as they do so. Avoiding such mistakes is one of the principal challenges that military intelligence analysts face.[16] Some such behavior will always occur. But experienced analysts should be able to filter much of the bad information received before it affects their databases. We can reflect the quality of order-of-battle analysts and database managers in our model by placing a *threshold* on the value of discrimination ratios that a Blue element will accept. For an element with the best analytic capability possible, we would set the threshold at unity and accept only ratios equal to or less than one. As an element's analytic capability falls, we can allow the threshold to rise.[17]

We can expand Eq. (4.3) to show how a series of new data affect the quality of information in a database. Suppose we use Eq. (4.3) to express the quality of information in an inference based on nonempirical information and all empirical data available.[18] Decompose those data into two sets, $\mathbf{x}_1$ and $\mathbf{x}_2$. Then Eq. (4.3) tells us that:

[16] A large literature addresses this problem in many settings, not just those relevant to military intelligence. The classic study of this behavior in military intelligence can be found in Wohlstetter, 1962.

[17] This approach implicitly assumes that increasing analytic capability affects only a Blue element's ability to avoid the biggest Type II errors (accepting the wrong hypothesis). Increasing capability is probably more likely to correct Type II than Type I errors (rejecting the true hypothesis); in fact, increasing capability should reduce both. If better information were available, a more refined approach to this phenomenon might be warranted.

[18] This includes all data collected to date. We properly degrade the quality of old data to reflect its level of quality at the moment of processing.

$$\frac{1 - p(A_t \mid x_1, x_2)}{p(A_t \mid x_1, x_2)}$$

$$= \left(\frac{1 - p(A_t)}{p(A_t)} \right) \left(\frac{p(x_1, x_2 \mid O_t)}{p(x_1, x_2 \mid A_t)} \right) \qquad (4.4a)$$

$$= \left(\frac{1 - p(A_t)}{p(A_t)} \right) \left(\frac{p(x_1 \mid O_t)}{p(x_1 \mid A_t)} \right) \left(\frac{p(x_2 \mid x_1, O_t)}{p(x_2 \mid x_1, A_t)} \right) \qquad (4.4b)$$

$$= \left(\frac{1 - p(A_t \mid x_1)}{p(A_t \mid x_1)} \right) \left(\frac{p(x_2 \mid x_1, O_t}{p(x_2 \mid x_1, A_t)} \right) \qquad (4.4c)$$

Equation (4.4a) simply restates Eq. (4.3), decomposing x into x_1 and x_2. Equation (4.4b) decomposes the discrimination ratio in Eq. (4.4a) into one based solely on x_1 and one based on x_2, given that x_1 is available. If we apply Eq. (4.3) to the first two ratios on the right in Eq. (4.4b), we get the expression in Eq. (4.4c). This last expression is fully analogous to Eq. (4.3), but now the dependence of the a priori odds ratio on empirical data is explicit. This illustrates why we can refer to Eq. (4.3) as an updating formula; it allows us to incorporate one set of data, x_1, into an a posteriori odds ratio that then becomes an a priori odds ratio that we update with new data, x_2. We can repeat this process as many times as necessary to incorporate many new datasets. As Eq. (4.4a) shows, the product of this sequential process is one final a posteriori odds ratio based on a discrimination ratio of two joint probability density functions. And Eq. (4.4b) shows that we can partition this discrimination ratio into a product of many discrimination ratios, each focused on a new dataset.

The decomposition of the total discrimination ratio into ratios associated with each new dataset reflects potential dependencies among these datasets. If these sets are independent, the decomposition is especially clean; in this case

$$p(x_1, \ldots, x_n \mid A_t) = p(x_1 \mid A_t) \times \ldots \times p(x_n \mid A_t)$$

and we can write Eq. (4.4b) as

$$OR(x_1, \ldots, x_n) = OR(0)\, DR(x_1) \times \ldots \times DR(x_n)\,, \qquad (4.5)$$

where $OR(x_1, \ldots, x_n)$ is the a posteriori odds ratio, $OR(0)$ is the a priori odds ratio, and each $DR(x_i)$ is a discrimination ratio based only on x_i. This decomposition dramatically simplifies the use of Bayes' Theorem. We exploit this simplicity to simulate how an intelligence system transforms a series of new empirical data into intelligence products and, more specifically, how the quality associated with these new data affect the quality of the intelligence products developed.

Given the level of aggregation we pursue in this simulation, a simple device of this kind looks extremely attractive. But we must keep it in perspective.

Expert systems that use a Bayesian approach almost always assume independence.[19] Without this assumption, the need for data to quantify each conditional probability becomes so great that the confidence we can place in each estimate fades away. Expert systems that assume independence tend to perform well relative to the alternatives, even when clear dependencies are present and not modeled in the systems. This may suggest that, in complex systems, the importance of statistical dependency to parts of the system need not make dependency so important when we view the system as a whole. In fact, to exploit the opportunities presented by statistical dependency, a complex system may have to focus more and higher-quality information in one place for a decision than the system can focus on a regular basis; selected and highly visible cases where a system's exploitation of a dependency made a difference are not characteristic of the system's normal capability. Our simulation is not an expert system, and it is not meant to predict behavior. But if these factors help explain the success of expert systems that assume statistical independence, they would suggest that we could safely make a similar assumption.

Two important sources of statistical dependence could create difficulties in our model. First, suppose a collector introduces the same information into the system twice. It is not reasonable to suggest that the intelligence system could get anything from the second set that it had not extracted from the first. This is simply an extreme case of a situation in which two similar collectors gather similar information from the battlefield at about the same time. We cannot say that the second collector adds much information that the first collector had not already gathered. Taken together, measures from the two will tend to wash out the effects of measurement error associated with each of them, but the value added by this "cross-checking" process is limited. If we attempt to model intelligence development in which such collection occurs often, assuming independence will tend to overstate the

[19] For a useful survey of the literature, see Ramsey et al., 1986.

quality of information that the system produces on unit-attributes relevant to the collectors in question.[20]

Second, dependencies exist in intelligence fusion and can play an important part in testing hypotheses. Suppose many Red units use radar type R_1, many others use radar type R_2, but only one uses both R_1 and R_2. A new sighting based on seeing a radar of type R_1 or R_2 will suggest little discriminatory power in the sighting. But successive sights of R_1 and R_2 in the same location or a single sighting of both together would identify the unit without question. That is, the discrimination ratio for the intersection of seeing R_1 and R_2 in one location is far smaller (better) than the product of the ratios for seeing R_1 and R_2 separately. Assuming independence will tend to understate the quality of information generated by joint sighting of both types of radars in a unit.

This example reflects a specific application of a general principle in collection management: When looking for a particular item on the battlefield, look for the set of indicators that jointly distinguish that item, even if individually they are of little use. The difficulty of collecting information on different indicators simultaneously vitiates the power of this principle in practice. Perhaps for similar reasons, an assumption of independence has proven useful in simulations in other settings (for example, medical diagnosis) where such dependencies are important. But the contribution of such dependencies to inferences about Red behavior can be important when experienced intelligence teams can cue sensors quickly and should prove useful when the Guardrail Common Sensor facilitates simultaneous collection of ELINT and COMINT. We must be alert to this possibility and make adjustments for it, as necessary.[21]

Given the simplicity that it allows and the success others have had using an assumption of independence in similar settings, we use it here as well. In doing so, we must keep in mind the potential difficulties it presents and be alert to circumstances in which they unduly color our analysis. Under this assumption, we can use Eq. (4.3) to update information in the following ways:

[20] A variation on this problem actually exacerbates problems in real intelligence systems. Attempts to achieve redundancy in communication often allow one new piece of information to enter an intelligence system in more than one form. A Blue element, receiving the information from two different sources, can and occasionally does take the second arrival as confirmation for the first. We do not allow such errors to occur in our simulation, even though they do occur in some real intelligence systems.

[21] For example, if an intelligence system routinely uses Common Sensor data on COMINT and ELINT to exploit dependencies between these disciplines, we should assure that the value of these dependencies is reflected in the discrimination ratios we use to characterize new sightings from Common Sensor data streams.

1. The a priori odds ratio represents the quality of information in a database when an interim intelligence product arrives.
2. The Blue element responsible for this database decides whether to accept the interim intelligence product based on the size of its discrimination ratio. The better the capability of the Blue element, the smaller the discrimination ratio has to be for the element to accept this product; the very best Blue element accepts only products with discrimination ratios of less than one.
3. Once a Blue element accepts an interim intelligence product and uses it to update its database, the discrimination ratio associated with the intelligence product shows how new information embedded in this intelligence product will affect the quality of information in the database. Assuming independence effectively means that the way the information in a new intelligence product affects the quality of information in a database does not depend on where information embodied in the new product or the database originally came from.[22]
4. The a posteriori odds ratio produced when the Blue element uses new information to update its database, properly degraded, serves as the basis for the a priori odds ratio in the database when the next intelligence product with new information arrives.

Note that very small and very large values of odds ratios may present a problem here. If we do not allow discrimination ratios to take values of zero, the processes above will never reduce an odds ratio to zero. But if an odds ratio approaches zero, it will be difficult for any new discrimination ratio to raise its value. Similarly, if an odds ratio becomes large, it may take an inordinately long time to raise the quality of information to a reasonable level. To avoid these difficulties, we reserve the possibility of setting maximum and minimum values for odds ratios. Bayesian logic provides some basis for choosing a maximum value. For example, a diffuse subjective probability distribution for a unit-attribute that can take only five values would presumably assign a probability of 0.2 to each category, suggesting a maximum odds ratio of 4 (0.8/0.2). More generally, for a unit-attribute that can take n categorical values, the maximum odds ratio is (n − 1). Appropriate minimum values for unit-attributes with categorical values

[22] To help avoid the first source of dependence mentioned above, we do not allow information from any sighting of a unit-attribute to affect the database in any Blue element more than once. Hence, we know that if new information has affected an odds ratio once, it will not affect it again.

are unclear; minimum and maximum values of location and speed are also unclear a priori. We expect experience with the model to suggest empirically useful levels to use.[23]

HOW THE MODEL DEGRADES AND UPDATES INFORMATION

Assuming independence allows us to associate a specific discrimination ratio with each new sighting of a unit-attribute on the battlefield. Our simulation of the quality of intelligence effectively moves this discrimination ratio through the intelligence system as intelligence products that reflect information from this new sighting flow through the system. As it moves, our simulation degrades it and uses it to upgrade older information. Figure 4 illustrates this process for new information on a particular sighting in an intelligence system with one collector, one processor, and one user.

The new sighting occurs at t_0. The model sets the initial value of the discrimination ratio at that time. The new information leaves a collector at time t_1 and arrives at a processor for initial processing at time t_3. We use Eq. (4.1), with an appropriate value of D, to degrade the discrimination ratio over a period from t_0 to t_3. If the adjusted value of the ratio is below the processing threshold, we continue. If not, we discard this information and wait for the next piece of information. If we continue, the processor last received new information and updated its database at time t_2. We use Eq. (4.1) to degrade the information in the processor's database over a period from t_2 to t_3. With appropriately adjusted inputs, we use the Bayesian updating formula in Eq. (4.3) to update the processor's database. The processor sends this updated information to the user at t_4 and the user receives it at t_5. We use Eq. (4.1) to degrade the updated information from t_3 to t_5 and record the result to determine the quality of information that the user received.

Once a discrimination ratio from a sighting of a unit-attribute enters the system, it influences the quality of information about that unit-attribute for the remainder of the scenario. To see this, ask what happens to the next sighting that enters the intelligence system in Fig. 4.1. When information about it reaches the processor, information on the previous sighting (the one in the last example), appropriately upgraded, is present in the processor's database. The discrimination ratio from

[23] Such minima and maxima are unrelated to the threshold discussed above. Minima and maxima reflect *general* aspects of fusion *when it occurs*. The threshold discussed above reflects a *specific* Blue element's capability to discern the true information content of new data and to decide *whether to fuse it* with existing information in its database.

Time	Real-World Event	Simulation Event
t_0	Blue collector observes a unit-attribute on the battlefield.	Generate a new discrimination ratio for this unit-attribute.
t_1	Processor updates its database with information from a previous sighting.	Apply Eqs. (4.1) and (4.3) as shown below.
t_2	Collector sends new information to the processor.	None.
t_3	Processor receives new information and uses it to update its database.	1. Use Eq. (4.1) to degrade new information from t_0 to t_3. 2. Check value of discrimination ratio against threshold. If and only if it exceeds the threshold, continue. 3. Use Eq. (4.1) to degrade information in database from t_1 to t_3. 4. Use Eq. (4.3) to transform a priori odds ratio in database into an a posteriori odds ratio.
t_4	Processor sends updated information to the user.	None.
t_5	User receives new information.	1. Use Eq. (4.1) to degrade updated information from t_3 to t_5. 2. Record quality of information sent to user.

Fig. 4—Example of new information moving through an intelligence system

this previous observation influences the a priori odds ratio in the processor updated by information on this new sighting and information on every sighting that follows it. In fact, at any time, Eq. (4.5) tells us that the odds ratio in the processor is simply the product of its initial odds ratio, the discrimination ratios of all previous sightings accepted at the processor, and the cumulative degradation factor that applies for the scenario to date. The odds ratio that the user has at any time can be characterized in a similar way.

Given a set of initial discrimination ratios, the network associated with any intelligence system moves these discrimination ratios through the network until they influence the final products of the system.

INPUTS FROM VIC ON THE QUALITY OF INFORMATION

As explained above, we use the Army's VIC corps combat model to simulate combat on the deep battlefield and Blue collectors' gathering of information on this combat. VIC creates a set of information every time a collector "sees" a unit. We can use this information to calculate a discrimination ratio every time a collector sees a unit-attribute. As more refined analysis proceeds on the discriminatory power of collectors, more sophisticated formulas could be substituted for those simple ones presented here without affecting our basic approach.

VIC generates three numbers of particular interest. The first is the "probability of detection" associated with a sighting. In fact, it represents the fraction of the relevant portion of a unit that a collector sees during a particular sighting. For ELINT collectors, it is the fraction of the unit's radars detected. If COMINT, it is the fraction of the unit's radios detected. For IMINT, it is the fraction of the unit's vehicles and other major pieces of equipment. While such a concept is not really meaningful as a "probability of detection," it is quite useful as an indication of the quality of a sighting. The second number VIC generates is the "standard error" associated with a collector's detection of the unit's location. The third is a similar "standard error" associated with a collector's detection of the unit's speed. VIC uses these latter two numbers as inputs to a Kalman filter[24] that accumulates information from a series of sightings to calculate the parameters of subjective probability functions for the location and speed of each unit. We can use these standard errors as a basis for our own simulation before they enter VIC's Kalman filter.[25]

The VIC Probability of Detection and Discrimination Ratios

For unit-attributes with categorical values, VIC's probability of detection is the only value we can use to derive discrimination ratios. Suppose that, for any collector and relevant unit-attribute, a simple

[24] A Kalman filter is a statistical technique that uses individual additions to a sample of data to update estimates based on that sample. The VIC Kalman filter uses additional data on a unit's sighted location and velocity to estimate that unit's true location and velocity at a certain time.

[25] The subjective probability distributions that the VIC Kalman filter generates for location and speed reflect input from all collectors. We cannot use information from these distributions to show how changes in the use of collectors affect these distributions without running VIC under more than one set of assumptions. Our approach is designed to use VIC to provide one baseline run. Further, information on these distributions does not allow us to examine changes in an intelligence system other than changes in collectors. Therefore, we do not rely on information about the subjective probability distributions that VIC generates.

relationship existed between a VIC probability of detection and a discrimination ratio.[26] If we could specify the relationship between these two concepts at several distinct points, we could use these points to parameterize the simple relationship and hence to determine the values of a discrimination ratio that apply for all values of the VIC probability of detection.

Consider values of the VIC probability equal to zero, one, and some "typical" value. If a collector does not detect a unit, this "sighting" presumably has no discriminatory value. If a collector detects everything it could detect about a unit, discriminatory value associated with the sighting will vary by collector and attribute. For example, such a sighting by MTI would provide no information about a unit's name. It could provide highly accurate—but not perfect—information about its location. Counts of vehicles and observations on their activities could provide good information on effectiveness and activity. Such a sighting by COMINT is harder to interpret. It suggests that the Blue system intercepts and properly interprets all radio traffic for the duration of the sighting. Such a sighting could provide highly accurate information on a unit's name, type, and echelon, and good information about its location. Rarely, however, will such a sighting generate perfect information. A "perfect" sighting implied by a probability of detection of one is not equivalent to a zero discrimination ratio, and the value of such a perfect sighting can vary substantially by collector and attribute.

All of the statements above are qualitative. We must be able to state them quantitatively. For example, can we say that 90 percent of the units that display a certain pattern of vehicle movement observed by MTI are engaged in a forward march in the deep battlefield? If so, and MTI sightings of other activities have similar discriminatory power, we can assign a discrimination ratio of 0.11 (.1/.9) to MTI sightings of activities. This approach provides a structured way to consider such subjective assignments. Observation of peacetime exercises and broad-ranging, preliminary interviews with order-of-battle analysts suggest that sightings by sensors provide useful information on some attributes and none on others. In the latter case, we can proceed as though a sensor provided no new sighting. When sensors do provide useful information, appropriate values of the discrimination ratio can get as low as about 0.1. (Table 2 provides details for combinations of unit-attributes that take categorical values and collectors.) If a Blue

[26] Any particular collector provides information relevant only to certain unit-attributes. For example, JSTARS provides no information on unit name. We are concerned here only with establishing a relationship between a VIC probability of detection and discrimination ratio for unit-attributes relevant to a collector.

element's initial probability distribution for a unit-attribute is fairly diffuse, so that the associated odds ratio is one, a single sighting with a discrimination ratio of 0.1, not degraded by delays, could place 91 percent of subjective probability in the right category. With a similar initial condition, two independent sightings with discrimination ratios of 0.33 would have about the same effect.

What is the VIC probability of detection associated with the "typical" sighting of a collector and what discriminatory value does such a sighting offer? Like the perfect sighting, the value of a typical sighting will vary by collector and unit-attribute. Observation of peacetime exercises and preliminary discussions with analysts suggest, for sightings that yield useful information on an attribute, discrimination ratios can get as low as about 0.2. Two independent sightings of typical quality from appropriate collectors could change a fairly diffuse probability distribution to one in which over 90 percent of subjective probability fell in the correct category. This is consistent with the general rule of thumb used by order-of-battle analysts that they require two independent sources to place a piece of information in their databases.

Results based on preliminary interviews with order-of-battle analysts suggest that a striking regularity occurs across collectors and attributes. A given rise in the VIC probability of detection typically

Table 2

PARAMETER VALUES FOR THE RELATIONSHIP BETWEEN THE VIC PROBAILITY OF DETECTION AND THE DISCRIMINATION RATIO

	COMINT Internals		COMINT Externals		ELINT		IMINT MTI	
Attribute	a[a]	b	a	b	a	b	a	b
Name	0.1	−1.0	—[b]	—	—	—	—	—
Type	0.1	−0.8	0.2	−0.9	0.2	−0.9	0.5	−0.6
Echelon	0.1	−1.0	0.2	−0.9	0.2	−0.9	—	—
Effectiveness	0.3	−0.9	—	—	—	—	0.2	−0.6
Activity	0.2	−0.6	—	—	—	—	0.2	−0.9

[a] a is the discrimination ratio for a "perfect" sighting. b measures the percentage change in discrimination ratio for a 1 percent change in VIC's probability of detection.

[b] We do not use VIC data to generate sightings for the pairings of data sources and attributes marked by "—." For example, COMINT internal data sources are the only ones that generate sightings relevant to a unit's name.

has a larger effect on the discrimination ratio when the VIC probability is small than when it is large.[27] Figure 5 shows the VIC probability of detection (PD) on the abscissa and the discrimination ratio (DR) on the ordinate. It shows the three points discussed above as part of a functional relationship of the following form:

$$DR = (a)\ (PD^{b}) \qquad (4.6)$$

where a and b are constants. a is the value that the discrimination ratio takes for a perfect sighting. b shows the percentage change in the discrimination ratio for a 1 percent change in the VIC probability of detection and always takes an absolute value equal to or less than one. Table 2 reports reasonable values for a and b based on preliminary interviews with order-of-battle analysts. This table defines relationships for all unit-attributes except location and speed.

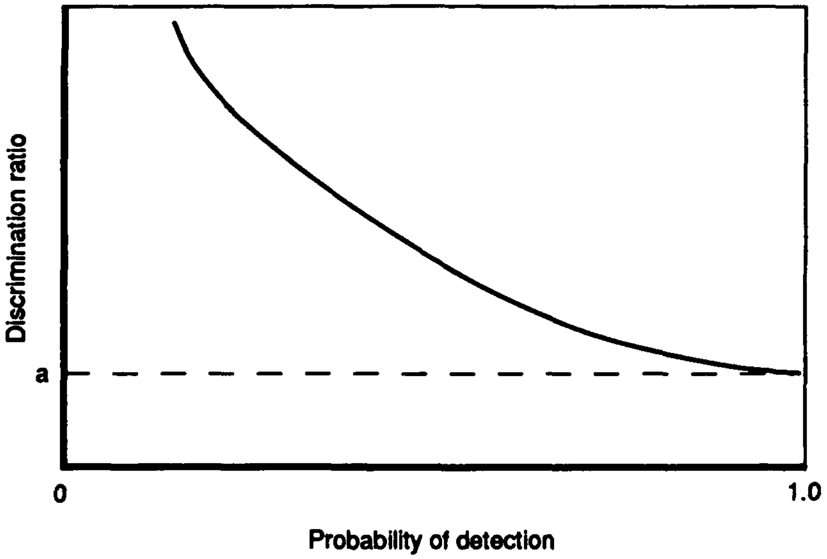


Fig. 5—Relationship between the VIC probability of detection and the discrimination ratio

[27] That is, the discrimination ratio appears to be concave in the VIC probability of detection for most collectors and attributes where a relationship exists.

VIC Standard Errors and Discrimination Ratios

For speed and location, VIC generates information on standard errors that gets us closer to the subjective probability distribution that we use to think about the quality of intelligence. We use our view of that distribution to derive a simple two-step relationship between VIC's standard error for a unit-attribute and a discrimination ratio. First we establish the relationship between the standard error of the subjective probability distribution for a unit-attribute and a corresponding odds ratio in our model. Then we use this relationship to transform the standard error that VIC reports for each sighting into a discrimination ratio for that sighting. For simplicity, we present this argument in terms of location; a completely analogous argument holds for speed.

Step 1. Define the region of the subjective probability distribution that we associate with the correct location as the one that lies within D meters of the actual location. Consider a normal subjective probability distribution for location. Project that distribution onto a line that passes through the actual location and the mean of the distribution so that we can think of accuracy about location in terms of a single standard error, SS, and a scalar mean on this line. Assume that the mean is displaced from the actual location by a distance that is proportional to the standard error. Hence, as precision increases, the standard error falls, concentrating subjective probability and bringing the central tendency of this concentration closer to the actual location. How to relate the standard error and bias in the mean is an open question. For simplicity, let us assume that the distance from the mean to the actual location equals the difference between the 75th and 50th percentiles of the normal subjective probability distribution on the line.[28] Then ask how the subjective probability that falls within D of the actual location, P, changes as the standard error changes. We can use a simple expression to define this relationship:[29]

[28] Here is a heuristic justification for this choice. Our real interest is in the absolute distance from the actual location to the mean. The subjective distribution tells us that we would place a 50 percent probability on that distance being larger than the difference between the 75th and 50th percentiles and a 50 percent chance that it would be smaller. Hence, choosing this distance gives us the median bias that is consistent with the distribution. This distance equals (0.6745) (SS).

[29] For selected values of SS, we can calculate values of the probability, P, that this distribution places within D of the correct location. These pseudodata confirm that a monotone relationship exists between SS and P. We use these pseudodata to estimate the relationship shown. It explains about 88 percent of the variation in the pseudodata for P. We chose the specific functional form shown in Eq. (4.7) simply because it is simple and fits the pseudodata well.

$$P = 1 - [(D/SS)+1]^{-3.4}. \qquad (4.7)$$

Step 2. How does SS relate to a new datum on the standard error of a sighting, SN? If SS is simply the product of a series of similar but independent observations that display a standard error of SN, then

$$SS = SN/(n^{.5}) \qquad (4.8)$$

where n is the number of observations in the series. In our model, the odds ratio is the product of a series of discrimination ratios and an a priori odds ratio. If the a priori subjective probability were fairly diffuse, so that the a priori odds ratio equaled one, and these observations were of similar quality, the discrimination ratios associated with them would be $((1 - P)/P)^{(1/n)}$. From Eqs. (4.7) and (4.8), we can see that

$$P/(1 - P) = [(n^{.5}) (D/SN) + 1]^{3.4} - 1$$

and that a discrimination ratio would take the value

$$DR = \{[(n^{.5}) (D/SN) + 1]^{3.4} - 1\}^{-1/n}. \qquad (4.9)$$

We calculated this value for different values of n and examined how the value of the enhancement increment varies in response to variations in D/SN. A value of n = 2 yielded values that made the most sense. The fact that order-of-battle analysts tend to seek two observations to confirm a location provides a check on this procedure. We can use a simple expression to capture the relationship in Eq. (4.9) for n = 2 :[30]

$$DR = (0.25) (D/SN)^{-1.3}. \qquad (4.10)$$

This is essentially the same functional form that Eq. (4.6) provides for transforming VIC probabilities of detection into discrimination ratios. We can easily provide values of the discrimination ratio appropriate for any circular region we choose to use to define a region of correct location. We can make Eqs. (4.6) and (4.10) fully analogous by choosing values of D to define a single region for location and for speed. Let D equal 0.3 km for location and 0.5 km/h for speed. Then we can restate Eq. (4.10) as

$$DR = (a) (SN^{b}) \qquad (4.11)$$

[30] We fitted this expression using pseudodata from the calculations discussed in the text. This relationship explains 91 percent of the variation in pseudodata for the discrimination ratio.

where the appropriate values of a and b are shown in Table 3. We do not require separate values of a and b for each type of c:"ector. Equation (4.11) simply transforms the standard error that VIC' reports for each collection into a discrimination ratio that we can use for our own calculations. No longer does a have a simpler intuitive definition; b is now the percentage change in the discrimination ratio associated with a 1 percent change in the standard error.

In sum, we use simple formulas to transform data from VIC into discrimination ratios for our simulation. These functions are based on subjective judgments that could presumably be refined by more detailed analysis of each collector and its ability to provide discriminating information on each attribute. More sophisticated methods for transforming VIC data into discrimination ratios could easily be substituted for these without affecting the structure or operation of our simulation.

SUMMARY

We frame our approach to the quality of information in terms of subjective probability distributions for unit-attributes. The more subjective probability its distribution places on the actual value of a unit-attribute, the higher the quality of a Blue element's information on that unit-attribute. We use a particular measure of information quality, based on this probability, that facilitates our approach. It is an "odds ratio," the probability above, divided into its complement. For unit-attributes that take categorical values, we look at the subjective probability associated with the correct category. For unit-attributes that take continuous values—location and speed—we essentially create a category in the vicinity of the actual value and look at the subjective probability associated with values in this vicinity. While order-of-

Table 3

PARAMETER VALUES FOR THE RELATIONSHIP BETWEEN VIC STANDARD ERRORS AND THE DISCRIMINATION RATIO

	Parameter[a]	
Attribute	a	b
Location	1.2	1.3
Speed	0.6	1.3

[a] a and b are parameters in Eq. (4.11).

battle analysts do not think of information quality in these formal terms, these terms capture essential features of the way analysts view uncertainty and facilitate our development of a simple analytic framework for simulating changes in information quality.

The quality of information changes as time passes and as analysts use new information to update their databases. To update databases, analysts continually posit hypotheses about the behavior of Red units and their implications for Red intentions and use new information to test these hypotheses and posit new hypotheses. This process is too complex to model in detail. We aim to capture the central features, including the following: We want the quality of information developed by an intelligence system to increase as Red units behave more predictably; new, relevant, good information enters the system; and the quality of new information increases. We want the quality of information developed by an intelligence system to decrease as time passes during which the system receives no new information, and the system accepts new information that is of low quality or deceptive. Our goal is to develop a method for showing how individual elements of an intelligence system—collectors, processors, communication links, and users—influence these factors in the system.

We start our simulation by characterizing the quality of the new information that enters the intelligence system through collection. We use the Army's VIC corps combat model to determine when Blue intelligence receives new information on each Red unit and to characterize the initial quality of the information that Blue intelligence receives. We use VIC's measures of the "probability of detection" and the "standard error" associated with each unit sighting. Simple formulas transform these into a measure of a "discrimination ratio" for each unit-attribute. The discrimination ratio measures the Blue intelligence system's ability to use this new information to discriminate between the hypotheses that a unit-attribute takes its true value and that it takes some other value. The higher the system's ability to use this information to discriminate between these hypotheses, the lower the ratio.

Once information enters the intelligence system, processors incorporate it into a series of increasingly complete intelligence products that culminate in products the system provides to its final users. This takes time. We expect delays to occur on communication links and in processors. These delays will degrade information for two reasons. First, as time passes without new information, Blue intelligence elements lose confidence that their (typically implicit) models are appropriately tuned to infer the current status and behavior of Red units. Second, as time passes without new information, it becomes increasingly difficult for Blue elements to associate new data with the appropriate Red units. We reflect the combined effects of these factors in our simulation with a simple exponential decay function that increases at a constant percentage rate the odds ratios and discrimination ratios that the Blue system associates with a unit-attribute.

As new information enters the intelligence system, it can potentially offset this decay by giving Blue elements an increased ability to discriminate between the hypothesis that Red unit-attributes take their true values and the hypothesis that they do not. New information enters the intelligence system and becomes embodied in successively more complete intelligence products until it influences the quality of information given to the system's users. At each element in the Blue system, we model the contribution of such new information to an element's database in the following way. When an interim intelligence product that reflects new information on a Red unit-attribute arrives at a Blue element, we observe the Blue element's odds ratio for the unit-attribute. We observe the information content of the new information, defined by its discrimination ratio. Bayes' Theorem tells us that, if a Blue element incorporates this new information into its database, the resulting quality of the database is the product of the odds ratio when the information arrived and the discrimination ratio. We assume the Blue element accepts new information if its value lies below a threshold, which we set at one if the Blue element is highly effective and increase as the effectiveness of the Blue element falls.

Over the course of a scenario, this simulation transforms a time series of information from VIC on probabilities of detection and standard errors associated with Blue sightings of Red units into a time series of the odds ratio that characterizes the quality of information the Blue intelligence system sends a user on each Red unit-attribute. Such time series provide the basis for policy analysis.

V. A SIMPLE EXAMPLE OF SIMULATED INFORMATION FLOWS AND INFORMATION QUALITY

This section uses a simple numerical example to illustrate how the simplified corps intelligence system shown in Fig. 3 uses three sightings from the Army's VIC corps combat model to develop intelligence on two attributes of a Red unit.

VIC generates information each time a collector gathers information on a Red unit. We transform this information into measures of the discriminatory quality of information on individual Red unit-attributes.

INFORMATION QUALITY OF VIC SIGHTINGS

The intelligence system shown in Fig. 3 includes three collectors:

- Guardrail Common Sensor COMINT
- Guardrail Commnn Sensor ELINT
- JSTARS MTI.

For our purposes, the first collector generates one data stream on the internal content of radio communication (GRCS-COMINT-intl) and a second on the external characteristics of radio communication (GRCS-COMINT-extl). The second collector generates a data stream on the technical characteristics of radars (GRCS-ELINT). These two collectors fly on a common platform but need not observe activities in specific units at the same time. They need not even generate sightings on the same units. The third sensor uses radar to generate a data stream on the movement of vehicles (JSTARS-MTI).

VIC generates a single sighting for each of these three collectors of each Red unit that it "sees" each time the collector goes on station. Each time Common Sensor ELINT or JSTARS MTI sights a unit in VIC, VIC generates a probability of detection, a standard error for location, and a standard error for speed. Our simulation transforms these into discrimination ratios on eight unit-attributes. Each time Common Sensor COMINT sights a unit in VIC, our simulation transforms these three data from VIC into 16 discrimination ratios—eight unit-attributes for each of two data streams.

Consider a situation in which each collector is on station during a single hour and VIC generates a sighting from each collector on a single Red unit during that hour. Table 4 shows these sightings at times 0, 30, and 45. That is, the MTI sighting occurs first; 30 minutes later the COMINT sighting occurs, and 15 minutes later the ELINT sighting occurs.

Let us focus on information from these sightings about two unit-attributes, one that takes categorical values, "unit type," and one that takes continuous values, "location." When these three sightings occur, VIC generates two pieces of information relevant to these unit-attributes. Table 4 shows values for the "probability of detection" and the "standard error" for location, measured in kilometers. We apply Eq. (4.6) to transform the probability of detection into a discrimination ratio for each categorical unit-attribute. Table 4 shows the outcomes for unit type. For the values shown, the odds that the actual unit type generated the data collected range from 1.3:1 to 4.3:1, not particularly high.[1] We apply Eq. (4.11) to transform the standard error for location into a discrimination ratio for location, shown in Table 4. The odds that a location near the true location generated the data collected range from 1.3:1 to 17:1. These discrimination ratios show us the quality of information from these VIC sightings on unit type and location and initiate our simulation of how an intelligence system uses new data from the battlefield.

Table 4

QUALITY OF INFORMATION FROM VIC SIGHTINGS

			Information from VIC		Initial Discrimination Ratios	
Data Source Number	Name of Collector	Time of Sighting	Probability of Detection	Standard Error	Type	Location
x1	GRCS-COMINT-intl	30	0.35	0.4	0.2316	0.3646
x2	GRCS-COMINT-extl	30	0.35	0.4	0.5145	0.3646
x3	GRCS-ELINT	45	0.30	0.7	0.5910	0.7548
x4	JSTARS-MTI	0	0.50	0.1	0.7579	0.0601

[1] These odds are the inverses of the extreme discrimination ratios shown.

DELAYS IN COMMUNICATIONS AND PROCESSING

An intelligence system takes time to transform new information into final intelligence products, depending on how long it takes to move information on specific communication links and to incorporate information in intelligence products at specific processors. Our simulation accepts information on these specific delay times as an input. Table 5 presents a notional set of delay times that we use for the current simple example.

Given these delay times, we can determine when intelligence products that reflect these new data arrive at various points in the intelligence system, *if* a Blue element does not reject them as substandard at some point. Table 6 displays these times. The table states time in terms of minutes following the initial MTI sighting. It also indicates which of the four original data sources, from Table 4, is reflected in the

Table 5

NOTIONAL DELAY TIMES IN COMMUNICATION AND PROCESSING

Cause of Delay		Delay in Minutes for: Low Priority Information	Delay in Minutes for: High Priority Information
Delays on Communication Links			
From	To		
GRCS-COMINT-intl	talk-processor	30	5
GRCS-COMINT-extl	com-extl-processor	15	1
GRCS-ELINT	ELINT-processor	15	1
JSTARS-MTI	MTI-processor	1	1
com-extl-processor	signal-processor	120	10
ELINT-processor	signal-processor	90	10
talk-processor	ASPS-processor	180	20
signal-processor	ASPS-processor	180	1
MTI-processor	ASPS-processor	180	1
ASPS-processor	corps-commander	360	15
ASPS-processor	arty-commander	540	45
MTI-processor	arty-commander	30	15
Delays Within Processors			
talk-processor		45	10
com-extl-processor		10	5
ELINT-processor		15	5
MTI-processor		5	1
signal-processor		5	5
ASPS-processor		120	15

product that arrives at each point. The arrival times in Table 6 essentially define how information flows through the intelligence system.

Note two important characteristics of arrival times at the corps commander's staff in Table 7. First, the intelligence products arriving at the commander's staff and embodying new data do not arrive in order of generation. We would also expect this outcome in a more complete simulation. Second, data generated over a 45-minute period yield final products that arrive over a period of about three hours. This has important implications that we cannot reflect here directly. If the collection pattern shown here is typical for any hour of a scenario, we would expect information from about 12 sightings to reach the commander during the three-hour period shown. Some would come

Table 6

ARRIVAL TIMES OF INTELLIGENCE PRODUCTS THAT REFLECT NEW DATA

At[a] Time	Results Based on Data Source Number	Arrive at
0	x4	JSTARS-MTI
1	x4	MTI-processor
30	x1	GRCS-COMINT-intl
30	x2	GRCS-COMINT-extl
45	x2	com-extl-processor
45	x3	GRCS-ELINT
60	x1	talk-processor
60	x3	ELINT-processor
160	x3	signal-processor
175	x2	signal-processor
186	x4	ASPS-processor
285	x1	ASPS-processor
345	x3	ASPS-processor
360	x2	ASPS-processor
666	x4	corps-commander
765	x1	corps-commander
825	x3	corps-commander
840	x2	corps-commander
846	x4	arty-commander
945	x1	arty-commander
1005	x3	arty-commander
1020	x2	arty-commander

[a] Time is stated minutes after the initial MTI sighting.

from collection earlier than those we show here, others would come from later collections. That is, the data we show here do not maintain their time ordering as they affect intelligence development in the system, nor do data from adjacent time periods, and they would tend to become intermingled with products we show here by the time they reached the corps commander. For the purposes of this illustration, we ignore these other sources of new data; we should not forget, however, that the actual simulation is somewhat more complex than this simple example.

INFORMATION QUALITY IN THE CORPS COMMANDER'S DATABASE

Intelligence products reflecting newly collected data eventually reach a database of particular interest to us, that of the corps commander. They affect the quality of the database that he and his staff maintain and use to support decisions. We can use information on (1) the initial discrimination ratios for each new data source, (2) delays from collection to receipt by the commander of products based on each new data source, (3) decay rates for each unit-attribute, and (4) odds ratios in the commander's database for each unit-attribute when the first product based on one of these data sources arrives, to calculate their effects on the quality of information in the commander's database.

We can get discrimination ratios from Table 5 and delay times from Table 7. For simplicity, in this example, we assume that the odds ratio for each unit-attribute equals 0.1 when the first product arrives. In a full simulation, this odds ratio would be an output of earlier calculations based on the quality of earlier information received. The only factor we do not have is a decay rate for each unit-attribute.

Table 7

QUALITY OF CORPS COMMANDER'S INFORMATION ON UNIT TYPE

Original Source of Information	Arrival Time	A Priori Odds Ratio	Decayed Discrimination Ratio	A Posteriori Odds Ratio
JSTARS-MTI	666	0.1000	(1.5371)	0.1000
GRCS-COMINT-intl	765	.1111	0.5054	.05614
GRCS-ELINT	825	.05984	1.3528	.08095
GRCS-COMINT-extl	840	.08225	1.2158	.10000

Choosing Decay Rates for This Example

Our goal in choosing decay rates is to create a reasonable level of information quality in the baseline intelligence system that we can use to judge the performance of an alternative intelligence system relative to the baseline. Assume that new information arriving at the corps commander's staff on each unit-attribute is just sufficient to offset the effect of information decay during the period of arrival; the intelligence system is in steady state for information about each unit-attribute.

We can state this condition in terms of our example in the following way. Let D be the decay rate for a unit-attribute. Let DR_i be the initial discrimination ratio for the ith data source that reaches the commander in some intelligence product. Let OR_0 be the odds ratio for the commander's database when the first product reflecting these data sources reaches the commander. Let OR_i be the odds ratio immediately following incorporation of an intelligence product based on the ith data source. Let TD_i be the delay time between collection and arrival at the commander for the ith data source. And let TO_i be the delay time between arrival of products based on the ith and (i + 1)th data sources.

Applying Eqs. (4.1) and (4.3) to the arrival of the product based on the first data source yields

$$OR_1 = (OR_0)(DR_1)\exp(D)(TD_1) \tag{5.1}$$

Applying Eqs. (4.1) and (4.3) to the arrival of the product based on the second data source yields

$$\begin{aligned} OR_2 &= OR_1\exp(D)(TO_1)(DR_2)\exp(D)(TD_2) \\ &= (OR_0)(DR_1)(DR_2)\exp[(D)(TD_1 + TD_2 + TO_1)] \,. \end{aligned} \tag{5.2}$$

If intelligence products based on all data sources ultimately arrive and are accepted, this process yields

$$\begin{aligned} OR_4/OR_0 &= DR_1 DR_2 DR_3 DR_4 \\ &\exp[D(TD_1 + TD_2 + TD_3 + TD_4 + TO_1 + TO_2 + TO_3)] \end{aligned} \tag{5.3}$$

To achieve the steady state we seek, we require a decay rate that sets the expression on the right to unity. That is, we seek

$$D = -(\Sigma \ln DR_i) / (\Sigma TD_i + \Sigma TO_i) \tag{5.4}$$

where the summations are defined over the data sources that actually affect the commander's database.

In our actual simulation, we do not explicitly apply Eq. (5.4). We use it here only to achieve a steady state. The quality of intelligence can rise and fall in a baseline intelligence system. We use thiε kind of logic to choose decay rates that generate an appropriate level of quality in the baseline intelligence system.

Calculating the Level of Quality in the Commander's Database

We are now in a position to present the effect of new intelligence on the quality of the commander's database. Table 8 presents information on the level of quality for information about unit type. For each new piece of information, it shows the original source, the arrival time, and three numbers relevant to the Bayesian updating formula. The table implements Eq. (4.3) by multiplying the a priori odds ratio in the database when new information arrives by the decayed discrimination ratio to yield the a posteriori odds ratio for the database following acceptance of new information. The table places the discrimination ratio for new information in parentheses if it does not affect this database.

Based on the information on these four sightings and their arrival times, we use a decay rate of .001062 per minute. Degradation occurs over time, not because it is hard to keep track of a unit's type (it rarely changes during a scenario), but because it is difficult to keep track of the unit itself. Continuity helps assure that analysts continue to apply data relevant to type to the right unit.

Information based on MTI data would arrive 666 minutes (about 11 hours) after it was collected if it was of high enough quality to enhance the intelligence products that the system develops. As shown, however, its degraded discrimination ratio is large enough that it would probably be excluded from databases at some point in the fusion process and never reach the commander. Our model would recognize this by setting a threshold low enough to exclude this from calculations. For the purposes of this illustration, let us assume a threshold of 1.4. In Table 8, it has no effect on the commander's information. Much higher quality information arrives 99 minutes later based on COMINT internals collected 735 minutes (12 hours, 15 minutes) earlier. By this time, the quality of the commander's database has eroded 11 percent in the absence of new information. This new information updates the database, greatly increasing its quality. Information based on ELINT arrives 60 minutes later that was collected 780 minutes (13 hours) earlier. Its quality has eroded with time, but analysts do not catch this and allow it in the database. Together with the passage of time, this drives down the quality of the commander's information. Information based on the last data source in our example arrives 15 minutes later. Information based on COMINT externals, collected 810 minutes (13 hours, 30 minutes) earlier has also degraded in quality. Analysts do not catch this. Together with the passage of time without new information, this drives the level of quality in the database to its level when the first information would have arrived.

Table 8 presents similar information about the quality of information on location. We can trace changes in the quality of the commander's information on location through Table 8 in a similar way. For the four sightings in this table and their arrival times, the steady-state decay rate is .003347 per minute. This rate is much higher than that for unit type. As noted in Sec. IV, we would expect the decay rate to be higher for location than for type because location is such a volatile attribute. Hence, even though we derive this rate from the somewhat arbitrary data on these sightings, it is realistic to expect a high decay rate here. (For this reason, it is also realistic to expect higher initial discrimination ratios here than for unit type.)

In Table 8, information based on an MTI sighting 666 minutes earlier arrives and substantially upgrades the commander's database. Given the high decay rate and the time it takes to get information to the commander, this is the only information that remains useful many hours after it is collected. It is reasonable to expect that analysts would reject other data and simply let the quality of the commander's information fali as time passes without fresh empirical input. This occurs in the table until, by assumption, the quality of the commander's information on location returns to its starting value.

Table 8

QUALITY OF CORPS COMMANDER'S INFORMATION ON UNIT LOCATION

Original Source of Information	Arrival Time	A Priori Odds Ratio	Decayed Discrimination Ratio	A Posteriori Odds Ratio
JSTARS-MTI	666	0.10000	0.5585	0.05585
GRCS-COMINT-intl	765	0.07798	(4.2688)	0.07798
GRCS-ELINT	825	0.09510	(10.2739)	0.09510
GRCS-COMINT-extl	840	0.10000	(5.4869)	0.10000

The a posteriori odds ratios in Tables 7 and 8 provide the basis for data series on the quality of the commander's information on unit-attributes. Figure 6 shows time of arrival on the abscissa and the subjective probability that the commander places on the right values of unit type and location.[2] In a complete simulation, such time series might measure quality at many hundreds of points in time over the course of a scenario for each unit-attribute.

MEASURING THE EFFECT OF AN INCREMENTAL CHANGE

This numerical illustration is far too simple to use to calculate the effects of an incremental change in an intelligence system. But we can exploit its simplicity to see the channels through which an incremental change would affect our measures of information quality.

Consider, for example, a change that eliminated the availability of COMINT internals. This could result from changes in the intelligence system that eliminated the collection of data on internals, eliminated

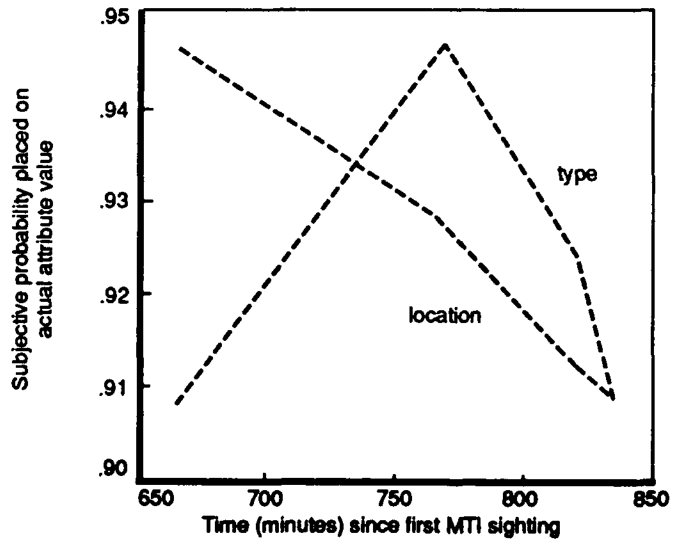


Fig. 6—Subjective probability that the commander associates correctly

[2] We derive these probability values (p) from values for odds ratios in Tables 7 and 8 (r) with the simple transformation, p = 1/(1 + r).

the availability or training of interpreters to listen to internals, deemphasized cryptographic analysis in a way that removed our ability to break Red codes and read their messages, or deemphasized internals in a way that sharply reduced connectivity with the rest of the system. In the extreme, we can represent any of these by removing the data stream on COMINT externals. What difference would this make?

In answering this, we must be careful. In this example, we assume a beginning odds ratio in the commander's database for each unit-attribute. Unless we adjust that, we can only model a loss of COMINT internals that occurs shortly before the time series in our example starts. That does not present a serious problem over the course of a scenario that runs for several days, but it seriously distorts the picture that comes from looking at one hour of collection. Again, our goal in this example is to understand how we use our approach, not to capture nuances of change in a real intelligence system.

From Tables 7 and 8, a loss of COMINT internals would remove a row from each table. For unit type, this would yield the results shown in Table 9. The one source of positive information enhancement is now gone and information quality falls steadily through the table. The loss of the row for COMINT internals in Table 8 has no effect on any other part of the table. Because analysts do not use COMINT internals to determine the location of the unit in this example, the loss of this information has no effect. In sum, our approach helps us determine that, for these very simple simulated data, losing COMINT internals would substantially affect the quality of the commander's information on unit type, but not that on unit location. It also provides a quantitative measure of what the effect would be.

We can think of other changes in an intelligence system in a similar way. A change in processing or communication time in a particular

Table 9

QUALITY OF CORPS COMMANDER'S INFORMATION ON UNIT TYPE IN THE ABSENCE OF COMINT INTERNALS

Original Source of Information	Arrival Time	A Priori Odds Ratio	Decayed Discrimination Ratio	A Posteriori Odds Ratio
JSTARS-MTI	666	0.1000	(1.5371)	0.1000
GRCS-ELINT	825	0.1184	1.3528	0.1602
GRCS-COMINT-extl	840	0.1627	1.2158	0.1979

part of the intelligence system could change the arrival times in Tables 7 and 8 for information based on data from different sources. Decay rates would remain the same as in the baseline. This would alter the degraded discrimination ratios in these tables and potentially change the value of information from different data sources in the commander's database.

A change in the processing of information from a particular collector could change its discriminatory value. We could represent this departure from the baseline intelligence system by adjusting parameter values in relationships that convert VIC data into discrimination ratios. Decay rates from the baseline would not change. Such a change would alter the adjusted discrimination ratios that enter Tables 7 and 8 and alter odds ratios in the commander's database.

A change in the analytic capability of the system could change the threshold used to reject poor-quality new information. Such a change could potentially be targeted within the intelligence system so that it affected only data flowing through that part of the system. This would tend to raise the discrimination ratios reaching the commander and have effects like those above.

We emphasize again that this illustration is too simple to give a detailed sense of how changes affect information quality. But its simplicity allows us to use it to trace the mechanics of how we model change in the simulation. In each case, we construct a baseline and then alter some part of it, inducing changes in the elements of Tables 7 and 8. These alter the time series we use to measure quality levels over time. Changes in these time series provide the basis for policy analysis.

SUMMARY

The example we offer here is a very simple one. We start by extracting specific data when collectors sight individual units in VIC. We transform these data into discrimination ratios that define the quality of information in these sightings relevant to the attributes of these units and the data streams generated by these collectors. Each discrimination ratio that we generate potentially initiates a series of events as the information associated with this ratio works its way through the network that defines the intelligence system. How rapidly information moves depends on exogenous assumptions about delay times on each communication link and in each processor. Once we know how fast information moves, we can examine the quality of information maintained in the commander's database. When data arrive at the commander's database, we degrade their quality to reflect the passage of time and ask whether their quality is higher enough for the intelligence system to incorporate them in this database. If it is, we degrade the database for the passage of time and then use Bayes' Theorem to calculate the effects of new information on it. We choose a degradation factor to achieve the pattern of quality that we would expect in the commander's database in a baseline case. We then use this factor to analyze departures from the base case that we can represent in terms of changes in the discrimination ratios associated with new information, delay times in the intelligence system, or thresholds used to determine which data reach the commander.

Although our example is simple, it conveys the essence of our approach. An actual simulation would use many more collectors and processors, a complete set of attributes, many more sightings by collectors, more realistic (and hence classified) assumptions about the quality of information from collectors and delay times, and hence more realistic baseline degradation factors. But at its heart, a full-blown simulation would simply execute the calculations illustrated here on a much larger scale.

VI. MODEL STRUCTURE AND IMPLEMENTATION

TERMINOLOGY AND OVERVIEW

This section presents the main data structures and process control flows within the model. For ease of reference, we call the computer implementation of this model PRO (for "intelligence PROpagation model"). To avoid confusion in the discussions within this section, we use "simulation" to represent the VIC ground truth simulation, and "model" to represent our computer model of an intelligence collection and fusion system.

The PRO model uses a battlefield simulation that is external to itself to generate two databases: (a) ground truth, a stream of readings of the location, ID, and other attributes of battlefield units at certain snapshots in time; and (b) a stream of sensor sightings. These sensor sightings later undergo a transformation process into what we call "pre-observations" that contain indications of the *quality* of (*not* the value of) time-stamped sensor readings of individual attributes of individual battlefield units. Later, another transformation step converts the pre-observations into normal observations[1] that flow among nodes in our model of a communication network.

For now, we are using the VIC battlefield simulation. In the future, other simulations may be used to provide this information to PRO.

Between VIC and PRO, we use a "VIC Postprocessor" program to map VIC outputs into the correct format for PRO inputs.[2] The simplified version of the data flows among these programs is shown in Figure 7.

An important characteristic of our system design lies in the fact that the two data flows from VIC are not dependent on any of the subsequent processing performed on those data flows. Consequently, each

[1] We are using the word *observation* not in the sense of a recorded measurement, but in the deeper sense of "a judgment or an inference from what one has observed." (Webster's *New Collegiate Dictionary*, 1979 edition.) The (pre-)observations derived from VIC may result from the action of a collection of sensors or represent other aggregated data, although we invariably extract these data before they enter one of the Kalman filters built into VIC to perform fusion.

[2] This is not the postprocessor that the VIC simulation uses to organize and present its output. It is a special program we have written to link VIC and PRO.

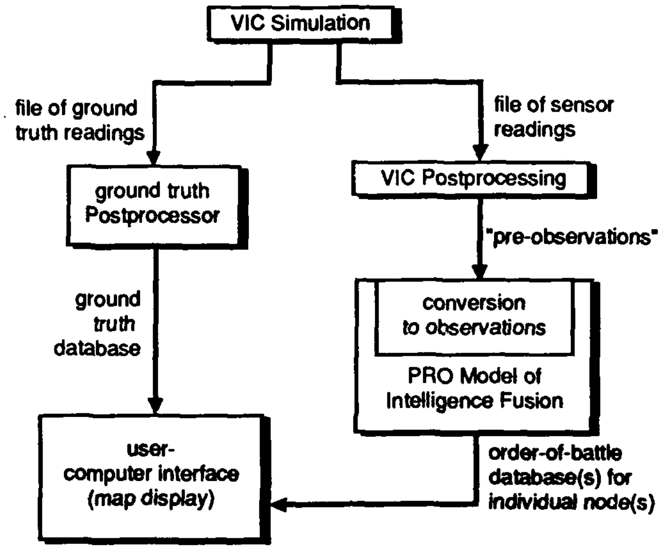

VIC Simulation
file of ground truth readings
file of sensor readings
ground truth Postprocessor
VIC Postprocessing
"pre-observations"
ground truth database
conversion to observations
PRO Model of Intelligence Fusion
user-computer interface (map display)
order-of-battle database(s) for individual node(s)

Fig. 7—Schematic of top-level PRO data flow

data flow is routed to a data file from a run of the VIC simulation; then many differing runs of the PRO model can be performed (for example, varying some parameter for sensitivity analysis) using those same VIC output data files. Because of VIC's large size and processing requirements, this results in major savings of time and project resources.

MODIFICATIONS TO VIC AND ITS OUTPUT DATA FLOWS

At its simplest, VIC may be considered an "engine" that moves both enemy and friendly *units* around on a battlefield. It also simulates the collection activities of various sensors.

As one of its standard outputs, the VIC simulation can produce a ground truth data stream. We have additionally inserted a set of PRINT statements into VIC to generate a stream of sensor sightings of unit attributes resulting from the actions of sensors.

Ground Truth Data Stream

A portion of a typical ground truth data stream generated by VIC during its operation is shown in Figure 8.[3] At certain time steps (we have chosen four-hour and 1/2-hour time steps for the recording of these ground truth "snapshots" in various of our VIC runs), a set of lines or records are emitted into the ground truth data file, providing current values for several attributes (unit ID, location, mass of unit, etc.) for *every* unit being simulated. Since this is a standard VIC report, it contains some data that are not relevant to our model (such as kill rate, loss rate, decon status, distance to the FLOT). The unit attributes that we use in this report are described below.

This "raw" ground truth data file is fed into a ground truth postprocessor[4] that: (1) extracts units of interest based on unit type and echelon,[5] and (2) transforms VIC-style unit-attributes into ones that are more readable and appropriate for subsequent PRO modeling. A typical output ground truth file resulting from ground truth postprocessing is shown in Figure 9.

We retain the VIC ground truth information for Blue units as part of this data file for context; the location and movement of Blue units on the map display screen in the user interface indicate the FLOT and aids in interpreting the movements of the Red units.

Unit attributes stored in the revised ground truth data file are the following. The file contains one record for each unit of interest at each time step.

Unit-ID A unique identifier for this unit, as provided by VIC. These are *not* the real identities of units, because of the sensitivity of these data. The unit-ID is used as an index to other attributes, identifying the unit and placing it within an organizational hierarchy. This is the same code as the NAME field within the raw VIC ground truth report.

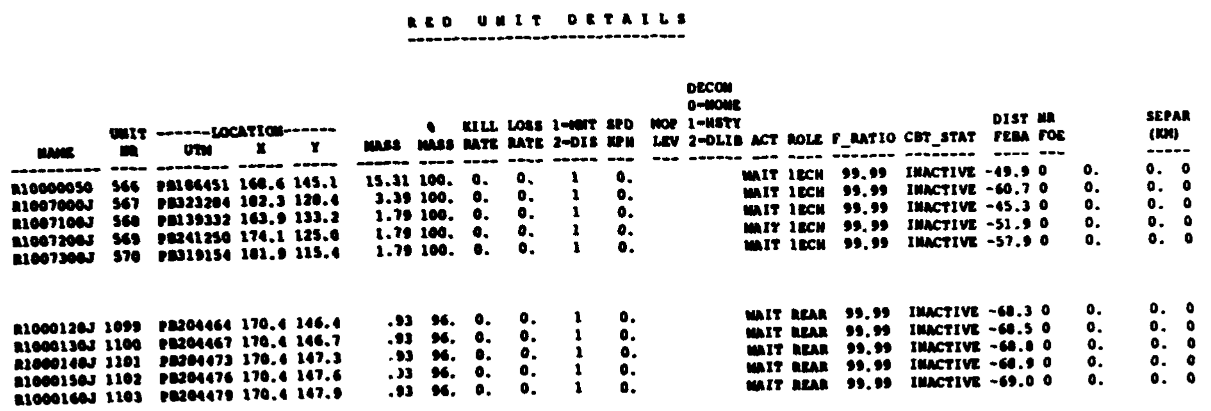
Examples of unit-IDs are:

B11113007
R11000050
R1409000B
R1114
R10171007
R140C4007

[3] The standard VIC output file called SS.HISTORY.FILE contains location and strength information for each unit at user-selected intervals during the simulation. A small portion of this file for one of our VIC runs is reproduced in Figure 8.

[4] Ground truth postprocessing is performed by a routine called "truth.prl," which we have written in PERL, a language in the public domain that operates under UNIX. It was written by Larry Wall of NASA/JPL and is useful for scanning text files, extracting information from them, and producing reports based on that information.

[5] Currently, unit types of interest are: tank, mechanized-infantry, infantry, cavalry, tube-artillery, headquarters, aviation-HQ, and artillery-HQ. Echelons of interest are: front, army, corps, division, brigade, regiment, and battalion.



```
                                             R E D   U N I T   D E T A I L S
                                             -------------------------------

                                                                           DECON
                                                                           0-NONE
             UNIT  ------LOCATION------              %   KILL LOSS 1-MNT SPD   MOP 1-HSTY                              DIST NR             SEPAR
   NAME       NR     UTM      X     Y      MASS   MASS RATE RATE 2-DIS KPH   LEV 2-DLIB ACT  ROLE F_RATIO CBT_STAT   FEBA FOE             (KM)
---------- ---- -------- ----- -----    ------ ---- ---- ---- ----- ---   --- ------ ---  ---- ------- --------   ---- ---             -----
R10000050   566  PB186451 168.6 145.1    15.31 100.  0.   0.    1    0.               WAIT 1ECH  99.99  INACTIVE -49.9 0      0.       0.  0
R1007000J   567  PB323204 182.3 120.4     3.39 100.  0.   0.    1    0.               WAIT 1ECH  99.99  INACTIVE -60.7 0      0.       0.  0
R1007100J   568  PB139332 163.9 133.2     1.79 100.  0.   0.    1    0.               WAIT 1ECH  99.99  INACTIVE -45.3 0      0.       0.  0
R1007200J   569  PB241250 174.1 125.0     1.79 100.  0.   0.    1    0.               WAIT 1ECH  99.99  INACTIVE -51.9 0      0.       0.  0
R1007300J   570  PB319154 181.9 115.4     1.79 100.  0.   0.    1    0.               WAIT 1ECH  99.99  INACTIVE -57.9 0      0.       0.  0


R1000120J 1099  PB204464 170.4 146.4      .93  96.  0.   0.    1    0.               WAIT REAR  99.99  INACTIVE -68.3 0      0.       0.  0
R1000130J 1100  PB204467 170.4 146.7      .93  96.  0.   0.    1    0.               WAIT REAR  99.99  INACTIVE -68.5 0      0.       0.  0
R1000140J 1101  PB204473 170.4 147.3      .93  96.  0.   0.    1    0.               WAIT REAR  99.99  INACTIVE -68.8 0      0.       0.  0
R1000150J 1102  PB204476 170.4 147.6      .93  96.  0.   0.    1    0.               WAIT REAR  99.99  INACTIVE -68.9 0      0.       0.  0
R1000160J 1103  PB204479 170.4 147.9      .93  96.  0.   0.    1    0.               WAIT REAR  99.99  INACTIVE -69.0 0      0.       0.  0
```

Fig. 8—Sample from VIC history file used as basis for ground truth

```
# Ground truth from file /v/jim/SIMD07, created 3 Jan 89.
\001\001\001\001
#                                                   AT TIME ( 0: 0: 0)
00 00:00
#Unit-ID   Side   Unit-type             Echelon     Activity                    Effective  Modelled Predict   Latitude  Longitude  Veloc-speed  Direction
#--------  -----  --------------------  ----------  --------------------------  ---------  -------- -------   --------  ---------  -----------  ---------
B11110007  Blue   artillery-HQ          brigade     inactive                        100.0  No       Yes         50.485      9.351         0.00  east
B11111007  Blue   tube-artillery        battalion   inactive                        100.0  No       Yes         50.738      9.713         0.00  east
B11112007  Blue   tube-artillery        battalion   inactive                        100.0  No       Yes         50.589      9.741         0.00  east
B11113007  Blue   tube-artillery        battalion   inactive                        100.0  No       Yes         50.504      9.704         0.00  east
R10000050  Red    headquarters          front       inactive                        100.0  Yes      Yes         50.752     10.584         0.00  east
R1007000J  Red    artillery-HQ          army        inactive                        100.0  Yes      Yes         50.626     10.770         0.00  east
R1008000J  Red    artillery-HQ          army        inactive                        100.0  Yes      Yes         50.806     10.889         0.00  east
R1009000J  Red    artillery-HQ          army        inactive                        100.0  Yes      Yes         50.715     10.465         0.00  east
R10100007  Red    artillery-HQ          division    inactive                        100.0  Yes      Yes         50.646     10.157         0.00  east
R10170007  Red    artillery-HQ          division    inactive                        100.0  Yes      Yes         50.634      9.966         0.00  east
R10174007  Red    tube-artillery        battalion   inactive                        100.0  Yes      Yes         50.463     10.034         0.00  east
R10180007  Red    artillery-HQ          division    inactive                        100.0  Yes      Yes         50.783     10.019         0.00  east
R10184007  Red    tube-artillery        battalion   inactive                        100.0  Yes      Yes         50.502     10.034         0.00  east
R10190007  Red    artillery-HQ          division    inactive                        100.0  Yes      Yes         50.725     10.031         0.00  east
```

Fig. 9—Sample from file resulting from ground truth postprocessing

We currently track data on 178 Red units with unique but artificial VIC names. We can easily adjust the postprocessor to track other units if appropriate.

Unit-side This indicates the side of the unit. Certain attributes of Blue units are stored within the ground truth database so that their positions may be displayed along with the perceived positions of the Red units. This attribute is used to display units of different sides in different colors. The value of this attribute is derived from the first letter of the unit-ID code.

Allowable values for unit-side are:

red	(indicates enemy unit)
blue	(indicates friendly unit)

Unit-type The generic type of the unit. This attribute allows certain processing to be performed for all units of a particular generic type. This information is derived from a lookup table within the ground truth postprocessor based on the unit-ID.

Allowable values for unit-type of primary interest in the current PRO model, with their corresponding VIC unit-type codes, are:

tank	[VIC: TNK]
mechanized-infantry	[VIC: MEC]
infantry	[VIC: INF]
cavalry	[VIC: CAV]
tube-artillery	[VIC: TUB]
headquarters	[VIC: HQ]
aviation-HQ	[VIC: HQV]
artillery-HQ	[VIC: HQA]

We can easily include other unit-types modeled by VIC if appropriate.

Unit-echelon This attribute indicates the echelon level of the unit. These echelon values are independent of unit-type. Again they are derived by lookup table from the unit-ID.

Allowable values for unit-echelon of primary interest in the current PRO model, with their corresponding VIC unit-echelon codes, are:

front	[VIC: FRT]
army	[VIC: ARM]
corps	[VIC: COR]
division	[VIC: DIV]

brigade	[VIC: BDE]
regiment	[VIC: RGT]
battalion	[VIC: BN]

If appropriate, we could add other unit-echelon codes modeled by VIC: battery, company, platoon, squadron, task-force, troop.

Unit-activity This attribute indicates the activity in which the unit is engaged at any point in time. This attribute is a rewriting of the VIC code for combat status (CBT_STAT) contained within the raw VIC ground truth output report. Allowable values for unit-activity in the current model, with their corresponding VIC unit-activity codes, are shown in Table 10.

Unit-effectiveness A real number from 0.0 to 100.0, indicating unit-effectiveness. The value 0.0 represents total unit-ineffectiveness; 100.0 represents total effectiveness relative to its level of effectiveness at the beginning of a scenario. This number is the "% MASS" value in the VIC raw ground truth report, indicating the present level of effectiveness of the unit.

Unit-modeled A boolean value (either Yes or No) representing whether this unit is to be modeled within PRO. (Note: regardless of this setting, the unit's behavior is still simulated within VIC.) This attribute permits a larger database of units to be represented in PRO, only some of which might be modeled in any particular run. In this manner, selections from the larger unit database may be made without major restructuring of the model. In PRO as currently constituted, all units whose unit-side is "Red" have a unit-modeled value of "Yes," and all Blue units have a value of "No."

Unit-predictable A boolean value representing whether the actions of this unit are predictable or not—that is, whether the actions are consistent with Blue's intelligence preparation of the battlefield (IPB). The default value is Yes.

Unit-lat A real number representing the latitude of the unit in decimal degrees (range −100.0 to 100.0). South of the equator is negative, north positive. This is derived from the location attributes of the unit in the VIC raw ground truth report.

Unit-lon A real number representing the longitude of the unit in decimal degrees (range −180.0 to 180.0). West of Greenwich is negative, east positive. Derived from the location attributes of the unit in the VIC raw ground truth report.

Table 10

ALLOWABLE VALUES FOR UNIT-ACTIVITY[a]

PRO Activity Value	Corresponding VIC
ADA-acquiring	ADA ON
ADA-not-acquiring	ADA OFF
advance-unopposed	*ADVANCE
artillery-in-place	*ART STAT
delay	DELAY
engineer-HQ-in-place	EN HQ S
engineer-at-base-and-ready	EN READY
engineer-at-minefield	EN AT MF
engineer-moving-to-HQ	EN TO HQ
engineer-moving-to-base	
engineer-post-task	
engineer-to-minefield	EN TO MF
frontal-attack	*FRT ATK
frontal-defense	*FRT DEF
helicopter-at-base-ready	HC READY
helicopter-at-base	HC BA ST
helicopter-in-postflight	HC POFLT
helicopter-in-preflight	HC PRFLT
helicopter-on-station	HC ON ST
helicopter-to-base	
helicopter-to-station	
HC-TO-BO	
inactive	*INACTIVE
movement-to-contact	*CONTACT
pursue	*PURSUE
reinforcing	*REINF
supply-unit-active	SUPPLY
withdraw-unopposed	
withdraw-opposed	

[a] VIC activity codes prefaced with an asterisk are the activities we expect to see for the Red units being modeled.

Unit-veloc-speed A real number representing the speed of the unit in kilometers/hr. The speed is computed from the change in location of the unit from the previous time-step (for example, four hours ago) to the current one within the VIC raw ground truth report.

Unit-veloc-direction The general direction of movement of the unit. If unit-veloc-speed is 0, this attribute's value has no meaning and is ignored. This value is also computed from the change in location of the unit from the previous time-step to the current one.

Allowable values for unit-veloc-direction in the current model are:

north
south
east
west

After its postprocessing, ground truth consists of a collection of unit records at each time stamp. This collection contains one record for every unit of interest (both Blue and Red) giving the values of the above unit-attributes at that time stamp. This file can then be used as input to an interactive display of the locations (or other attributes) of units as a function of time as a scenario unfolds.

Sensor Sightings Data Stream from VIC

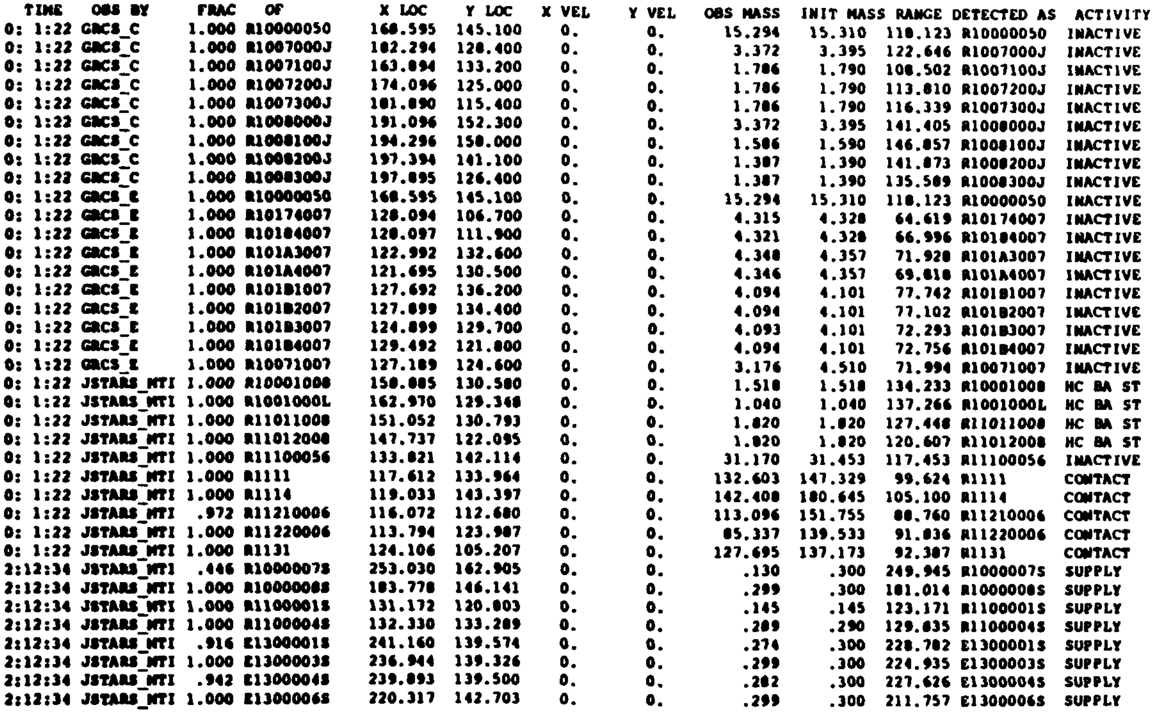
The second data stream is generated from PRINT statements inserted into VIC.[6] It consists of individual sensor sightings made at various (simulated) dates/times. At present, we have chosen to extract from VIC the observations made by the four categories of sensors described in Sec. III:

GRCS-COMINT-intl
GRCS-COMINT-extl
GRCS-ELINT
JSTARS-MTI

In the current version of the model, we take sensor readings labeled GRCS-C in VIC and duplicate them, providing identical outputs from the sensors we call GRCS-COMINT-intl and CRCS-COMINT-extl. The reason for this duplication and relabeling is so that these sensor sightings can be routed through two different communication and processing paths in our model.

An extract of a file of raw sensor sightings emitted from our PRINT statement inserted into VIC is shown in Fig. 10. Several of the attributes reported there are not currently being used; others are transformed into more useful sensor observation readings for the PRO model. That transformation and the mapping between the readings in Fig. 10 and the messages we call "pre-observations" are made by a VIC postprocessing step.

[6] Sensors in VIC are handled in the Fusion Intelligence (FI.SIM) module. The routine FI.CARRY.OUT.OBSERVATION calls the routine FI.CREATE.AN.OBSERVATION to produce sightings of individual units. As soon as VIC creates the sighting, we write out a record with the relevant data. This prefusion sighting has not yet gone through the Kalman filtering process (FI.PERFORM.KALMAN.ESTIMATION) used to fuse location and velocity data from the sightings.



```
  TIME   OBS BY      FRAC   OF            X LOC    Y LOC   X VEL     Y VEL   OBS MASS  INIT MASS RANGE DETECTED AS  ACTIVITY
0: 1:22 GRCS_C      1.000 R10000050     168.595  145.100    0.        0.      15.294    15.310  118.123 R10000050  INACTIVE
0: 1:22 GRCS_C      1.000 R1007000J     182.294  128.400    0.        0.       3.372     3.395  122.646 R1007000J  INACTIVE
0: 1:22 GRCS_C      1.000 R1007100J     163.894  133.200    0.        0.       1.786     1.790  108.502 R1007100J  INACTIVE
0: 1:22 GRCS_C      1.000 R1007200J     174.096  125.000    0.        0.       1.786     1.790  113.810 R1007200J  INACTIVE
0: 1:22 GRCS_C      1.000 R1007300J     181.890  115.400    0.        0.       1.786     1.790  116.339 R1007300J  INACTIVE
0: 1:22 GRCS_C      1.000 R1008000J     191.096  152.300    0.        0.       3.372     3.395  141.405 R1008000J  INACTIVE
0: 1:22 GRCS_C      1.000 R1008100J     194.296  158.000    0.        0.       1.586     1.590  146.857 R1008100J  INACTIVE
0: 1:22 GRCS_C      1.000 R1008200J     197.394  141.100    0.        0.       1.387     1.390  141.873 R1008200J  INACTIVE
0: 1:22 GRCS_C      1.000 R1008300J     197.895  126.400    0.        0.       1.387     1.390  135.589 R1008300J  INACTIVE
0: 1:22 GRCS_E      1.000 R10000050     168.595  145.100    0.        0.      15.294    15.310  118.123 R10000050  INACTIVE
0: 1:22 GRCS_E      1.000 R10174007     128.094  106.700    0.        0.       4.315     4.328   64.619 R10174007  INACTIVE
0: 1:22 GRCS_E      1.000 R10184007     128.097  111.900    0.        0.       4.321     4.328   66.996 R10184007  INACTIVE
0: 1:22 GRCS_E      1.000 R101A3007     122.992  132.600    0.        0.       4.348     4.357   71.928 R101A3007  INACTIVE
0: 1:22 GRCS_E      1.000 R101A4007     121.695  130.500    0.        0.       4.346     4.357   69.818 R101A4007  INACTIVE
0: 1:22 GRCS_E      1.000 R101B1007     127.692  136.200    0.        0.       4.094     4.101   77.742 R101B1007  INACTIVE
0: 1:22 GRCS_E      1.000 R101B2007     127.899  134.400    0.        0.       4.094     4.101   77.102 R101B2007  INACTIVE
0: 1:22 GRCS_E      1.000 R101B3007     124.899  129.700    0.        0.       4.093     4.101   72.293 R101B3007  INACTIVE
0: 1:22 GRCS_E      1.000 R101B4007     129.492  121.800    0.        0.       4.094     4.101   72.756 R101B4007  INACTIVE
0: 1:22 GRCS_E      1.000 R10071007     127.189  124.600    0.        0.       3.176     4.510   71.994 R10071007  INACTIVE
0: 1:22 JSTARS_MTI 1.000 R10001008     158.885  130.580    0.        0.       1.518     1.518  134.233 R10001008  HC BA ST
0: 1:22 JSTARS_MTI 1.000 R1001000L     162.970  129.348    0.        0.       1.040     1.040  137.266 R1001000L  HC BA ST
0: 1:22 JSTARS_MTI 1.000 R11011008     151.052  130.793    0.        0.       1.820     1.820  127.448 R11011008  HC BA ST
0: 1:22 JSTARS_MTI 1.000 R11012008     147.737  122.095    0.        0.       1.820     1.820  120.607 R11012008  HC BA ST
0: 1:22 JSTARS_MTI 1.000 R11100056     133.821  142.114    0.        0.      31.170    31.453  117.453 R11100056  INACTIVE
0: 1:22 JSTARS_MTI 1.000 R1111         117.612  133.964    0.        0.     132.603   147.329   99.624 R1111      CONTACT
0: 1:22 JSTARS_MTI 1.000 R1114         119.033  143.397    0.        0.     142.408   180.645  105.100 R1114      CONTACT
0: 1:22 JSTARS_MTI  .972 R11210006     116.072  112.680    0.        0.     113.096   151.755   88.760 R11210006  CONTACT
0: 1:22 JSTARS_MTI 1.000 R11220006     113.794  123.987    0.        0.      85.337   139.533   91.836 R11220006  CONTACT
0: 1:22 JSTARS_MTI 1.000 R1131         124.106  105.207    0.        0.     127.695   137.173   92.387 R1131      CONTACT
2:12:34 JSTARS_MTI  .446 R1000007S     253.030  162.905    0.        0.        .130      .300  249.945 R1000007S  SUPPLY
2:12:34 JSTARS_MTI 1.000 R1000008S     183.778  146.141    0.        0.        .299      .300  181.014 R1000008S  SUPPLY
2:12:34 JSTARS_MTI 1.000 R1100001S     131.172  120.803    0.        0.        .145      .145  123.171 R1100001S  SUPPLY
2:12:34 JSTARS_MTI 1.000 R1100004S     132.330  133.289    0.        0.        .289      .290  129.835 R1100004S  SUPPLY
2:12:34 JSTARS_MTI  .916 E1300001S     241.160  139.574    0.        0.        .274      .300  228.782 E1300001S  SUPPLY
2:12:34 JSTARS_MTI 1.000 E1300003S     236.944  139.326    0.        0.        .299      .300  224.935 E1300003S  SUPPLY
2:12:34 JSTARS_MTI  .942 E1300004S     239.893  139.500    0.        0.        .282      .300  227.626 E1300004S  SUPPLY
2:12:34 JSTARS_MTI 1.000 E1300006S     220.317  142.703    0.        0.        .299      .300  211.757 E1300006S  SUPPLY
```

Fig. 10—Sample from file of raw sensor sightings from VIC

VIC Postprocessing

The PRO model works by modeling the flow of a set of "messages" among communication paths within an intelligence fusion communication net being studied. Most messages contain information about the enhancement (or degradation) in the *quality* of a processing node's information about an *attribute* of a (Red) *unit* at a particular simulated date/time, caused by a sensor sighting, or by subsequent fusion activities of processing nodes. The purpose of the VIC postprocessing step is to transform the information in the raw sensor sightings (as illustrated in Fig. 10) into messages in the standard form for subsequent PRO processing. The result of this postprocessing is a stream of what we call "pre-observations," because they are still not quite in the form of a normal PRO observation.

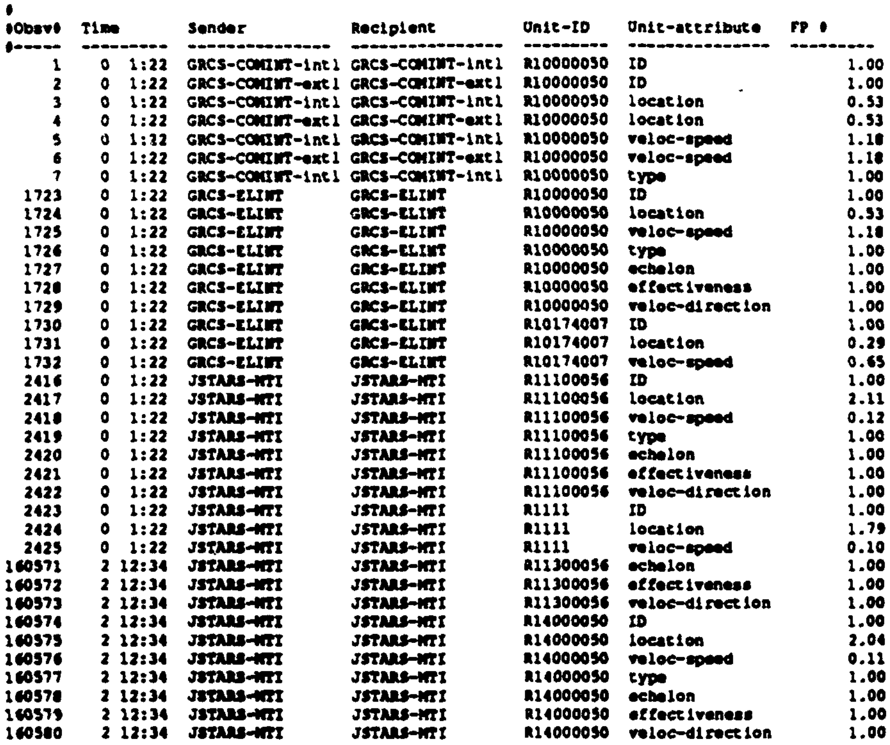
An extract from a typical file of pre-observations emerging from VIC postprocessing is shown in Fig. 11. Each pre-observation is given a unique observation number, a (simulated) date/time stamp, the name of the sensor making the observation (recorded as both Sender and Recipient), the unit-ID of the unit observed, the name of the attribute observed, and an "FP #" (floating point number) representing the quality of the sighting. Each line describes only a single unit-attribute combination as observed by a single sensor, whereas the records in the raw sensor sightings (Fig. 11) contain information on several unit-attributes. One record in the raw sensor sighting file therefore generates multiple records in the resulting pre-observations file.

The date/time stamp, the names of the Sender and Recipient nodes representing the type of sensor sighting, and the unit-ID of the unit in the pre-observations file are taken straightforwardly from the raw sensor readings file record. Each raw sighting record generates seven pre-observations with the above common attributes, each having a distinct unit-attribute – FP# pair as shown in Table 11. The FP # reading associated with each unit-attribute in the pre-observations is an indicator of the quality of the sensor sighting. This number is the basis for the calculation of an "enhancement increment" associated with this message when it becomes transformed from a pre-observation into a normal observation in the PRO model. That transformation is performed as follows: Recall from Eq. (4.6) that for *categorical* unit-attributes,

$$\mathrm{DR} = \mathrm{a} \times (\mathrm{probability\_of\_detection})^{\mathrm{b}}, \tag{6.1}$$

where DR is the discrimination ratio.

We use the FP # derived above as a surrogate for probability of detection (which is in turn a surrogate for the quality of the



```
#
#Obsv#  Time       Sender           Recipient         Unit-ID    Unit-attribute   FP #
#-----  ---------  ---------------  ----------------  ---------  ---------------  ---------
     1    0  1:22  GRCS-COMINT-intl GRCS-COMINT-intl  R10000050  ID                    1.00
     2    0  1:22  GRCS-COMINT-extl GRCS-COMINT-extl  R10000050  ID                    1.00
     3    0  1:22  GRCS-COMINT-intl GRCS-COMINT-intl  R10000050  location              0.53
     4    0  1:22  GRCS-COMINT-extl GRCS-COMINT-extl  R10000050  location              0.53
     5    0  1:22  GRCS-COMINT-intl GRCS-COMINT-intl  R10000050  veloc-speed           1.18
     6    0  1:22  GRCS-COMINT-extl GRCS-COMINT-extl  R10000050  veloc-speed           1.18
     7    0  1:22  GRCS-COMINT-intl GRCS-COMINT-intl  R10000050  type                  1.00
  1723    0  1:22  GRCS-ELINT       GRCS-ELINT        R10000050  ID                    1.00
  1724    0  1:22  GRCS-ELINT       GRCS-ELINT        R10000050  location              0.53
  1725    0  1:22  GRCS-ELINT       GRCS-ELINT        R10000050  veloc-speed           1.18
  1726    0  1:22  GRCS-ELINT       GRCS-ELINT        R10000050  type                  1.00
  1727    0  1:22  GRCS-ELINT       GRCS-ELINT        R10000050  echelon               1.00
  1728    0  1:22  GRCS-ELINT       GRCS-ELINT        R10000050  effectiveness         1.00
  1729    0  1:22  GRCS-ELINT       GRCS-ELINT        R10000050  veloc-direction       1.00
  1730    0  1:22  GRCS-ELINT       GRCS-ELINT        R10174007  ID                    1.00
  1731    0  1:22  GRCS-ELINT       GRCS-ELINT        R10174007  location              0.29
  1732    0  1:22  GRCS-ELINT       GRCS-ELINT        R10174007  veloc-speed           0.65
  2416    0  1:22  JSTARS-MTI       JSTARS-MTI        R11100056  ID                    1.00
  2417    0  1:22  JSTARS-MTI       JSTARS-MTI        R11100056  location              2.11
  2418    0  1:22  JSTARS-MTI       JSTARS-MTI        R11100056  veloc-speed           0.12
  2419    0  1:22  JSTARS-MTI       JSTARS-MTI        R11100056  type                  1.00
  2420    0  1:22  JSTARS-MTI       JSTARS-MTI        R11100056  echelon               1.00
  2421    0  1:22  JSTARS-MTI       JSTARS-MTI        R11100056  effectiveness         1.00
  2422    0  1:22  JSTARS-MTI       JSTARS-MTI        R11100056  veloc-direction       1.00
  2423    0  1:22  JSTARS-MTI       JSTARS-MTI        R1111      ID                    1.00
  2424    0  1:22  JSTARS-MTI       JSTARS-MTI        R1111      location              1.79
  2425    0  1:22  JSTARS-MTI       JSTARS-MTI        R1111      veloc-speed           0.10
160571    2 12:34  JSTARS-MTI       JSTARS-MTI        R11300056  echelon               1.00
160572    2 12:34  JSTARS-MTI       JSTARS-MTI        R11300056  effectiveness         1.00
160573    2 12:34  JSTARS-MTI       JSTARS-MTI        R11300056  veloc-direction       1.00
160574    2 12:34  JSTARS-MTI       JSTARS-MTI        R14000050  ID                    1.00
160575    2 12:34  JSTARS-MTI       JSTARS-MTI        R14000050  location              2.04
160576    2 12:34  JSTARS-MTI       JSTARS-MTI        R14000050  veloc-speed           0.11
160577    2 12:34  JSTARS-MTI       JSTARS-MTI        R14000050  type                  1.00
160578    2 12:34  JSTARS-MTI       JSTARS-MTI        R14000050  echelon               1.00
160579    2 12:34  JSTARS-MTI       JSTARS-MTI        R14000050  effectiveness         1.00
160580    2 12:34  JSTARS-MTI       JSTARS-MTI        R14000050  veloc-direction       1.00
```

Fig. 11—Sample from file of pre-observations

observation) and use values for parameters a,b in the above equation that are dependent on both the type of sensor being used and the unit-attribute being detected by that sensor, as discussed in Sec. IV.

From Eq. (4.5), the discrimination ratio is multiplied by the a priori odds ratio to obtain the a posteriori odds ratio, given a discrimination ratio for a particular sighting, i:

$$\text{odds\_ratio}_i = \text{odds\_ratio}_{i-1} \times \text{DR}_i \qquad (6.2)$$

and from Eq. (4.1), we have a corresponding equation for the decay of information over a time interval from t0 to t1:

$$\text{odds\_ratio}_{t1} = \text{odds\_ratio}_{t0} \times \exp[-D(t1 - t0)] \qquad (6.3)$$

Table 11

PRE-OBSERVATIONS IN THE RAW SIGHTING FILE

Unit-attribute	Corresponding FP #
ID	- FRAC in raw sensor sightings file, which is fraction of actual current unit mass that was detected by the sensor
location	- (.0045 × RANGE) for GRCS-COMINT and GRCS-ELINT[a] - (.018 × RANGE) for JSTARS-MTI, where RANGE is the range reading in the raw sensor sighting file, representing distance from the sensor to (center of mass of) the observed unit
veloc-speed	- (.01 × RANGE) for GRCS-COMINT and GRCS-ELINT; - (.001 × RANGE) for JSTARS-MTI
type	- FRAC in raw sensor sightings file
echelon	- FRAC in raw sensor sightings file
effectiveness	- FRAC in raw sensor sightings file
veloc-direction	- FRAC in raw sensor sightings file

[a] VIC uses a product of this form to estimate the standard error of location associated with a sighting. The products shown for veloc-speed have an analogous interpretation. The numbers in the text were chosen more or less arbitrar:

As time intervals pass and several observations are received by a particular observer (node), repeated applications of the above equations yield a generalization of Eq. (6.3):

$$\text{odds\_ratio}_{\text{current}} = \text{odds\_ratio}_{\text{original}} \times \text{DR}_1 \times \ldots \times \text{DR}_n \times \exp[-\text{D}(\text{TD}_1 + \ldots + \text{TD}_m)] \qquad (6.4a)$$

where the TD_k are time intervals of various kinds (for example, communication times, processing times) that sum to the total time difference between "current" time and "original" time.

From Eq. (6.4a), the only effect of the original odds ratio is as a multiplicative constant on the final answer. Also the effect of an intelligence collection and fusion system is to multiply those original odds by a factor, and for the purposes of evaluating alternative intelligence collection and fusion systems, *all we are concerned with is that factor:*

$$\frac{\text{odds\_ratio}_{\text{current}}}{\text{odds\_ratio}_{\text{original}}} =$$

$$DR_1 \times \ldots \times DR_n \times \exp[-D(TD_1 + \ldots + TD_m)] \qquad (6.4b)$$

If this factor—call it the "intelligence effectiveness factor"—is greater than one, the quality of the commander's information about a particular unit-attribute at the current time is poorer than it was at the original time; if equal to one, the qualities are the same; and if less than one, the quality of his information has increased from the original to the current time.[7]

PRO implements Eqs. (6.1) and (6.2) in logarithmic form. We prefer a logarithmic form for four reasons:

- Repeated computations in PRO Eqs. (6.1) and (6.2) to obtain a posteriori odds for the quality of a unit-attribute are faster using the logarithmic form of the equations.
- It is simpler to present the operations in PRO using addition and multiplication instead of multiplication and exponentiation, respectively.
- Many people find that it is natural to think of very large and very small probabilities on an informal logarithmic scale.
- In other areas where multiplicative relationships describe the underlying phenomenon, the use of a logarithmic scale with decibel units has proven useful to elucidate the phenomenon.

Users will not always find it easiest to use the output of PRO in a logarithmic form. PRO provides a flexible environment in which a user can present outputs for final display or analysis as logarithms, odds ratios, subjective probabilities, or whatever other form the user finds most appropriate for a particular application.[8]

[7] PRO allows the user to enter values for $\text{odds\_ratio}_{\text{original}}$ and to multiply them by the expression in Eq. (6.4b) to calculate values of $\text{odds\_ratio}_{\text{current}}$. However, a user need not determine meaningful values of $\text{odds\_ratio}_{\text{original}}$. To use PRO, a user will be interested in the value of the expression in Eq. (6.4b) in two cases: the baseline case and a case following an incremental change in the baseline. He can use the ratio of values for the expression in Eq. (6.4b) to measure the effect of the incremental change in question. Because $\text{odds\_ratio}_{\text{original}}$ takes the same value in both cases, this ratio of values does not depend on values of $\text{odds\_ratio}_{\text{original}}$.

[8] For example, a user's real interest may be in "targetability." If targetability is defined as knowing the location of a unit's elements within, say, 500 meters, a user can fairly easily transform the output of PRO into a measure that states the subjective probability that the system places on a unit's elements being within 500 meters of the loca-

Taking the base 2 logarithm of both sides of Eq. (6.2), dividing both sides by $\text{odds\_ratio}_{\text{original}}$, and multiplying by −10 to obtain a decibel scale and switch the sign of the effect,[9] we obtain:

$$-10 \times \log_2\left(\frac{\text{odds\_ratio}_i}{\text{odds\_ratio}_{\text{original}}}\right) =$$

$$-10 \times \log_2\left(\frac{\text{odds\_ratio}_{i-1}}{\text{odds\_ratio}_{\text{original}}}\right) - 10 \times \log_2(\text{DR}_i) \qquad (6.5a)$$

We call the value of the term:

$$-10 \times \log_2\left(\frac{\text{odds\_ratio}_i}{\text{odds\_ratio}_{\text{original}}}\right)$$

the "enhancement" of the quality of the information caused by the receipt of observations 1 through i, and we call the term:

$$-10 \times \log_2(\text{DR}_i)\text{ , or equivalently from Eq. (6.2) :}$$

$$-10 \times \log_2\left(\frac{\text{odds\_ratio}_i}{\text{odds\_ratio}_{i-1}}\right)$$

the "enhancement increment" caused by observation i. In these terms, Eq. (6.5a) can be rephrased as:

$$\text{enhancement}_i = \text{enhancement}_{i-1} + \text{enhancement\_increment}_i \; (6.5b)$$

The values of enhancement and enhancement increment are in decibels. Since the enhancement value of unit-attribute information at a particular node measures the change in the quality of the information about that unit-attribute at that node after a set of observations and a time interval have passed, the initial enhancement value for all nodes and unit-attribute combinations is, by definition, zero.

As a discrimination ratio DR_i is received representing a particular sensor observation, it is converted within the PRO model to an enhancement increment term and added to the existing value of the

tions designated in the intelligence system. Alternatively, a user can transform PRO output into a statement that the intelligence system places, say, an 80 percent probability weight on the statement that it knows the location of a unit's elements to within x meters. Many other transformations are also possible. Although PRO software does not perform all such transformations, the output it produces contains the information required to generate such products with suitable manipulation.

[9] Switching the sign of the effect enhances the intuitive comprehension of the calculations. Positive numbers are "good"; negative numbers are "bad."

enhancement for that node-unit-attribute combination, as indicated by Eq. (6.5b).

Within the PRO model, an observation is converted to an enhancement increment by use of the formula:

$$\text{enhancement\_increment}_i = \text{max\_increment} + 10 \times \text{elasticity} \times \log_2 (\text{probability\_of\_detection}) \qquad (6.6)$$

where $\text{max\_increment} = -10 \times \log_2 (a)$ and $\text{elasticity} = -b$, given parameters a,b representing characteristics of the sensor as discussed in Sec. IV. Equation (6.6) follows from Eq. (6.1) and the definition of enhancement increment, above.

A similar derivation from Eq. (6.3) leads to the equation:

$$\text{enhancement}_{t\,+\,dt} = \text{enhancement}_t - (Kdt) \qquad (6.7)$$

representing the (exponential) decay of an enhancement value over period of time dt. In Eq. (6.7) the decay factor K is:

$$\frac{10 \times D}{\log_e(2)}$$

where D is the decay rate factor in Eq. (6.3). The decay coefficient, K, in Eq. (6.7) is in decibels/hour.

In Eq. (6.5b), if a computed enhancement increment for a unit-attribute observation is greater than zero, the observation increases the quality of information about that unit-attribute at a particular node; if zero, the quality is unchanged; and if less than zero, the quality decreases as a result of that observation.

All of Eqs. (6.1) through (6.7) are also used for computation of enhancement increments and their addition to existing enhancement terms for *continuous* attributes; in this case, the parameter "probability_of_detection" is replaced by the standard error calculated as the product of RANGE and a factor that depends on the attribute and sensor. VIC provides this appropriate value of RANGE, and the values of parameters a,b are derived as described in Sec. IV.

THE PRO MODEL

The PRO model follows the flow of observations of unit-attributes from sensors simulated within VIC through various intelligence processing nodes. We record not the data values associated with the observations, but rather an indicator of the quality of the observation, stored as an enhancement increment (EI). At any time in the PRO model, an enhancement value is recorded for each unit-attribute combination for each node being modeled; the enhancement value represents that node's belief in the correct value for the unit-attribute. These enhancement values are reduced as time passes to reflect the increasing staleness of the information, and they can be increased as new observations are received by a node providing higher-quality information.

The PRO model is programmed primarily in the RAND-ABEL language,[10] with some supplemental routines coded in the C language. RAND-ABEL was designed and implemented at RAND initially to support the advanced modeling required in the RAND Strategy Assessment System (RSAS).[11] In addition to providing succinct, readable program code, RAND-ABEL allows access to the Data Editor and graphics displays of RSAS, such as map displays of western Europe with overlays of military unit icons showing their location and movement. These major RSAS resources can provide interactive data updating during a model run and a graphic interactive interface to PRO with modest programming effort.

Our descriptions here of data and processing within the PRO model are informal but similar to the actual structures and algorithms within the RAND-ABEL code for PRO. This similarity should allow an interested reader to understand the code of the PRO model. Most of the data structures within PRO are represented by arrays; we use the notation N[x,y] to represent access to an array N, where x,y,... are the values of the indices of the information being sought in the array. The version of the RAND-ABEL language we are using at present allows only two-dimensional arrays. Our description of PRO uses n-dimensional arrays, where appropriate, for succinctness and clarity; within the model they are implemented as a collection of two-dimensional arrays.

In general, the PRO model is described in terms of the following concepts:

UNITS

"Things" on the battlefield, such as an armored division or a brigade. Units may be illusory, caused by wrong observations or wrong conclusions (see the appendix). The goal of the intelligence fusion process we are modeling is the determination of the

[10] See Shapiro et al., 1985 and 1988.

[11] For an introduction to the Rand Strategy Assessment System, see Davis, Bankes and Kahan, 1986. The basic functional elements of RSAS software are publicly releasable in a form called RAMP.

location, identification, and other attributes of (enemy) units on the battlefield. The measure of the success of the intelligence fusion process is the resulting enhancement or decay in the accuracy of a commander's perception of these enemy unit-attributes. Red units are modeled in PRO; some Blue units are passed along as ground truth elements so that they may be displayed to provide context for the model's outputs.

NODES

Persons, organizations, or other abstractions that receive observations, process them, or use them. Nodes are interconnected to form a communication network over which observations are transmitted. Certain nodes perform intelligence fusion on inputs received; others represent commanders who are end-user recipients of fused intelligence; others form the interface between PRO and VIC, acting as recipients of VIC intelligence observations and passing them on to one or more PRO nodes. Each node has associated "inbox" and processing time delays, and each link between nodes has an associated transmission time delay.

MESSAGES

Messages transmit observations from VIC into PRO, and between units modeled within PRO. They are also used to govern certain system operations, often to be performed at a future time (represented by the date/time stamp within the message).

ORDER OF BATTLE (OOB)

A snapshot of the state of the modeled world, giving some node's (for example, a commander's) perceptions at some point in time. A collection of these snapshots for some node is contained in that node's order of battle database. An OOB database may be compiled for any node at the discretion of the modelers.

ENHANCEMENT

The net increase (or decrease, if negative) of the quality of the intelligence (about a given attribute of a given unit, maintained at a given node) since the beginning of the simulation (normally the initiation of hostilities). Measured in decibels, it is proportional to the logarithm of the ratio of the initial to final odds ratio associated with the subjective probability that the attribute has the value known by the simulation to be the correct value. It is a measure of uncertainty about attributes of units.

Units

A set of (Red) units is being modeled. The behavior of these units on the battlefield as a function of time constitutes ground truth, and changes in the quality of Blue's perceptions regarding information about various attributes of these units constitutes our modeling of an intelligence collection and fusion system. We currently model 178 Red units; VIC includes as many as 500 that could be modeled. Within the ground truth component of PRO, each unit is described by the values associated with the set of its attributes. When representing a sensor observation, these unit-attributes contain not the corresponding data value, but an "enhancement increment" representing the change in the quality of a commander's perception of that unit-attribute at some point. (For the purposes of this EI, a single unit location EI is stored within the unit-lat attribute, and the unit-lon attribute is unused.) The unit-attributes we model are:

Unit-ID
Unit-side
Unit-type
Unit-echelon
Unit-activity
Unit-effectiveness
Unit-modeled
Unit-predictable
Unit-lat
Unit-lon
Unit-veloc-speed
Unit-veloc-direction

The PRO model also retains an additional attribute for each unit:

Unit-displayed A boolean value representing whether this unit is to be displayed on the interactive map display during model execution. In this manner, selections of modeled units may be made to keep the display uncluttered and to focus on the movements and attributes of certain units of interest.

If a unit is to be displayed, the icon representing the unit is derived from its unit-type and is currently the military map symbol representing one of the following unit-types:

armored
army_hq
artillery
artillery_hq
brigade_hq
cavalry
corps_hq
division_hq
front_hq
mechanized_infantry
regiment_hq

PRO can accept icons for additional unit-types. If a unit has a type with no specified icon, it is represented by a displayed point.

We are also interested in the hierarchical relationships among units. These relationships are not explicitly encoded within the unit-attributes listed above because this information can be extracted from the unit-ID codes provided by VIC. Examples of model behavior influenced by hierarchical relationships that we may include in future elaborations of PRO include the following types of rules:[12]

> The presence of three of five subordinate units of an army gives us an inference about the status of the army. If a PIR is "Find the 8th Tank Army," we might simply increase the priority for the components of the 8th Tank Army. We may also develop a figure of merit by asking how many of these components are identified over the course of the processing. The figure of merit in essence becomes a measure of our ability to make higher-level inferences about hierarchy.
>
> If attributes of three of five units of an army are identified to some degree of tolerance, the likelihood that attributes of the remaining units would be identified should increase. Hence, a rule about an information enhancement term (see below) for a unit-attribute might have as an antecedent some function of the enhancement term for related unit-attributes.

Nodes

The PRO model contains the representation of a communication network linking various "nodes." Nodes may have zero or more sources of incoming information, and zero or more output communication paths to other nodes. In general, nodes represent "sensors," "fusion shops," "commanders," "weapons systems," and other abstractions within the intelligence processing network. Nodes have a belief about a unit. It is the decisions of human nodes, the output of institutional nodes, and the observations of sensor nodes that we are modeling. A node is anything that is capable of forming hypotheses about units. The following attributes are recorded for each modeled node:

[12] One of the virtues of writing PRO in RAND-ABEL is the ability to introduce rules like these easily. As noted in Sec. IV, we do not believe it is possible to use such rules to model every last detail of fusion activities or even to attempt to capture key aspects of fusion. Rules like the examples shown here, however, can be used to highlight certain fusion activities. The rules shown here, for example, provide illustrations of some of the simplest "higher inferences" that analysts might draw from the sort of order of battle we track. In fact, because such simple inferences constitute an important part of order of battle development itself and can be represented by fairly easy rules, it may be quite appropriate to compare the ability of alternative intelligence systems to produce order of battle data in terms of these higher inferences rather than the basic order of battle data from VIC. More generally, such rules illustrate the kinds of rules that PRO can easily accommodate to highlight particular aspects of intelligence fusion.

Node-ID A unique identifier for this node. This identifier is used as an index into arrays holding the values of the attributes of a node.

In this model, valid attribute values for node-ID are:

GRCS-COMINT-intl
GRCS-COMINT-extl
GRCS-ELINT
JSTARS-MTI
talk-processor
com-extl-processor
ELINT-processor
MTI-processor
signal-processor
ASPS-processor
corps-commander
arty-commander

More values can be easily added.

Node-default-process-delay A pair of numbers (t-lo, t-hi) indicating the time delay, in minutes, a message encounters during processing by this node. These are delay times for low- and high-priority 12 unit/attribute combinations, as stored in the unit-priority array. In the current model, we use the processing delays shown in Table 12 as a function of type of node.

Node-maintain-OOB A boolean value indicating whether an OOB database is to be maintained for this node.

The values of the default delay times associated with a particular processing node may be changed by operation of the model. In future versions of PRO, we intend to implement a general mechanism for scheduling changes at a future time to any model parameter. This may be done by adding a new message type, called a "change-order," whose function is to effect a particular parameter change at the (simulated) date/time associated with the message.

Table 12

PROCESSING DELAY BY NODE[a]

	Processing Delay (minutes) for	
Node-ID	Low-Priority Units	High-Priority Units
talk-processor	45	10
COM-extl-processor	10	5
ELINT-processor	10	5
MTI-processor	5	1
signal-processor	5	5
ASPS-processor	120	15

[a] Processing delays are set to zero for nodes acting as originators and end-users of observations. The nodes listed in the table are all intermediate processing nodes.

Max_increment and Elasticity

To convert a probability of detection computed from VIC data into an enhancement increment using Eq. (6.3b), we must store values of max_increment and elasticity used in that equation. In the current model, we have chosen to make these values a function of node type and unit-attribute; that is, they depend on which sensor is being used and which unit-attribute is being detected. These values are stored in two arrays:

Max_increment[unit-attribute, node-ID]

Elasticity[unit-attribute, node-ID]

An example of the values for JSTARS-MTI sensor for each type of categorical unit-attribute is given below:

Node-ID	Unit-attribute	Max_increment	Elasticity
JSTARS-MTI	unit-type	$-10 \times \log_2(0.5)$	0.6
JSTARS-MTI	unit-activity	$-10 \times \log_2(0.2)$	0.9
JSTARS-MTI	unit-effectiveness	$-10 \times \log_2(0.2)$	0.6

A complete list of these settings for all sensor/unit-attribute combinations is given in Tables 2 and 3, where max_increment and elasticity are related to the factors a,b (respectively) by the formulas contained in the discussion following Eq. (6.6), above: max_increment = $-10 \times \log_2$ (a) and elasticity = –b.

Communication Network

The sightings emanating from (our abstracted and summarized model of) the collectors within VIC, after postprocessing to become what we call "pre-observations," go into a MESSAGES queue. After additional processing to turn their FP # into an enhancement increment for a particular unit-attribute-node combination, these observations are then passed to various processing nodes (for example, representing intelligence fusion centers), and eventually to end-user nodes. Some nodes can pass their processed observations to more than one subsequent processing node or end user. Nodes may be thought of as having three independent properties, each of which may or may not be present:

- Some nodes, which we call "collectors," may originate observations; (most of these observations really come from postprocessed VIC, but they "originate" into the fusion model through these nodes); other nodes might originate observations through other mechanisms exterior to the model;

- Some nodes, which we call "recorders," have order-of-battle databases periodically maintained to represent their perception of the battlefield at various simulated times. This feature is controlled by the node attribute "Node-maintain-OOB" listed above. (The only effect of calling a node a recorder is generation of an output file; this has no effect on the model itself.) It is useful for debugging purposes to call some nodes recorders (to generate an output file of their beliefs over time), even though those nodes would not normally hold beliefs regarding the intelligence product at that point;
- Some nodes, which we call "processing nodes" (correlators), receive messages, process them, and emit zero or more other messages to represent the transmission of their processed data to other nodes in the network.

A node can have all three properties, or any subset of them, or none of them. (A node having none of these properties would be disregarded; presumably it is a node that we wish to model at times, but it has been inactivated, at least temporarily.)

The particular communication network among nodes that we are currently modeling in PRO is shown in Fig. 12, which reproduces Fig. 3 for local reference. The PRO program is independent of this particular network; other communication networks linking a collection of nodes can be modeled merely by changing data tables within PRO.

The communication paths shown in Fig. 12 that are currently being modeled are labeled in that diagram as C1 . . . C12. There are some restrictions on unit-attributes that can move on each link in the intelligence system. Those restrictions, as itemized in the current PRO model, are listed in Table 13.

The data in Table 10 are stored within PRO in the following array:

Propagate_att[sender_node, receiver_node, unit_attribute]

For each active communication ("commo") link between specific nodes, we need to store the default transmission delay time for observation types of messages, for both low- and high-priority observations, for each type of unit-attribute. We can therefore think of the communication network at any moment as a "connection matrix" represented by a four-dimensional array:

Default_comm_delay[from_node, to_node, unit-attribute, lo-or-hi]

containing in each array location a number, d, indicating the amount of time delay in seconds that is the default initial value for observation

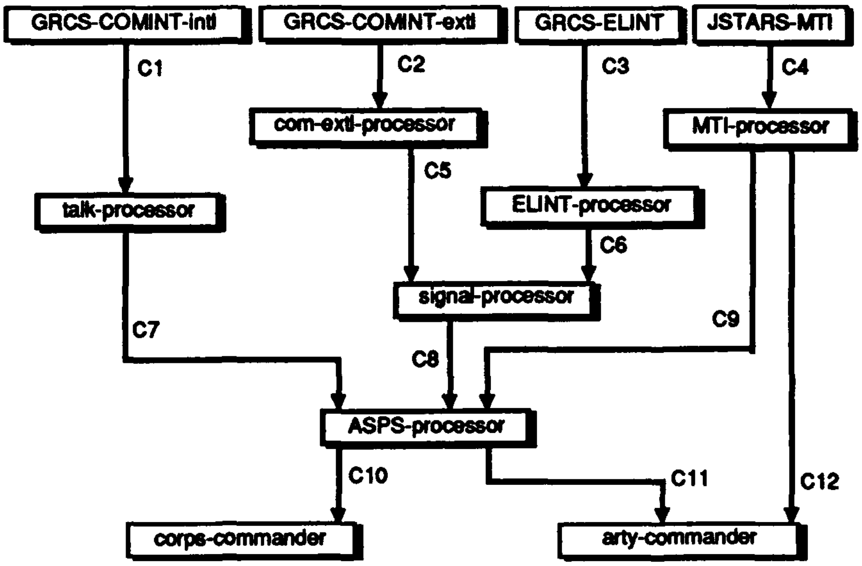


Fig. 12—Communication network currently modeled in PRO

Table 13

COMMUNICATION PATH RESTRICTIONS IN CURRENT PRO MODEL

Unit- Attribute	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
ID	yes	no	no	no	no	no	yes	no	no	yes	yes	no
type	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
echelon	yes	yes	yes	no	yes	yes	yes	yes	no	yes	yes	no
activity	yes	no	no	yes	no	no	yes	no	yes	yes	yes	no
effectiveness	yes	no	no	yes	no	no	yes	no	yes	yes	yes	no
location	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
veloc-speed	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
veloc-direction	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

transmissions over the link from the from_node to the to_node, for information regarding the specific unit-attribute (for example, a unit's ID, type, echelon, activity, effectiveness, location, speed, direction), and for low- or high-priority observations. Communication link delay times currently being used are shown in Table 5.[13]

The time from node1 to node2 may be different from the time from node2 to node1. If there were no communication link between node1 and node2, the value at the intersection has a special value (a very large number) indicating no link (an infinite delay).

During the operation of the model, rules could change which commo links are operational and the value of the time delay for any link.

Because the observation propagation network need not be a tree—indeed our initial network is not a tree—it is possible for observations based on the same pre-observation[14] to reach the same node by a different route. It would be inappropriate for two such observations to both increase the enhancement of the same node. From the structure of the specific communication net being modeled, shown above, one can in principle deduce the correct versions of propagation inhibition rules to avoid such double- (or higher) counting at a fusion node from a single pre-observation. In the general case, there are problems with doing this. For example, what if a given node receives information derived from the same pre-observation through two routes and the first to be received arrives with a smaller enhancement increment than the second? The proper thing to do would be to accept the first increment, but then when the second arrives, increase the node's enhancement by the difference between the second enhancement increment and the time-discounted first enhancement increment. At the present time, we avoid this complexity by incorporating a single highly *ad hoc* inhibition rule:

> An observation of location, received by ASPS-processor from MTI-processor, will not be propagated to ARTI-commander.

Similar rules can be written to handle this problem in other networks.

Messages

There is a queue called MESSAGES in which various messages are stored. (We use the word "queue" loosely here; the set of data iterns in MESSAGES is, at least notionally, kept sorted by date/time stamp as

[13] The Default_comm_delay array is sparse, unsymmetric, and more than half empty (since in our present model each link is only one way). Storage schemes might be used that save considerable computer memory; these will be implemented i– later PRO versions if space becomes a consideration.

[14] Recall from an earlier discussion that a pre-observation is the (effectiveness of the) reading of an individual unit-attribute by a sensor. They are derived from sensor readings emitted from the sensor simulator (in the present case, VIC).

items are removed from the queue and new items are placed into it as a result of processing.) The messages in the queue are of two types:

(1) "Observations," representing both (summarized) observations of (attributes of units on) the battlefield by sensors being simulated in VIC, and later fused (or otherwise processed) observations introduced into the queue by nodes representing intelligence fusion centers. Observations contain sporadic, selective quality information about certain unit-attribute data detected, with associated ID no. and time stamp identifying the collection event. Therefore, "snapshots" of the quality of unit-attribute readings, serial numbered and time stamped, emerge from (postprocessed) VIC for use in our model. Also, processing of an observation generates new observations.

(2) "Directives," administrative directives to the system causing a change in its behavior at the (modeled) date/time contained within the directive. Examples of directives are ones to halt the model's operation, to trigger computation and storage of certain nodes' "order of battle" perceptions, and to pause for human interaction with the model to take place.

As mentioned above, at a later time we expect to implement a third category of messages:

(3) "Change orders," telling the system to change some value in some data structure or table representing the model at the (modeled) date/time contained within the change order.

A message is a data object having the following attributes:

Mesg-type Indicates the type of message

Valid values for Mesg_type are:

observation
directive

Mesg-directive Indicates the specific directive being issued. The directive will be executed by PRO at the simulated date/time indicated within the message.

Valid values of Mesg-directive are:

read-truth	read the ground truth database
stop	stop model processing
log-OOB	write to a data file a "snapshot" the ground truth and order-of-

	battle databases for all nodes recording them
decay-enhancements	compute the time decay of enhancement values
update-all-unit-icons	redisplay all unit icons based on current enhancement values

Mesg-repeat-interval A time interval, in seconds. If nonzero, gives the frequency (in simulated seconds) at which to repeat this directive. Directives are repeated by incrementing their date/time by this repeat-interval value and reinserting them on the MESSAGES queue.

Mesg-observation-num An integer that is a unique ID given this observation by the VIC simulation. (Not used for directive-type messages.)

Mesg-time The modeled date/time at which this observation was made, or at which the directive it represents is to take effect.

Mesg-sender The Node-ID of the node that placed this message on the message queue.

Mesg-sender-type The Node-type of the node that placed this message on the message queue.

Mesg-recipient The Node-ID of the node that is the intended recipient of this message. If multiple nodes are to receive a message, a separate message is placed on the messages queue for each intended recipient. (Not used for directive-type messages.)

Mesg-unit-ID The unit-ID of the unit for which this is an observation. (Not used for directive-type messages.)

Mesg-unit-att The name of the unit-attribute for which this is an observation. (Not used for directive-type messages.)

Mesg-EI An "enhancement increment" for this unit-attribute observation. (Not used for directive-type messages.)

Note that messages relating to an observation DO NOT contain a data value for that observation (for example, the latitude or longitude of a unit observed). They only contain an "enhancement increment" representing the CHANGE in a commander's belief caused by this observation. This is a powerful feature of the PRO model design.

One of these data structures is generated for each unit-attribute pair observed by any collector during the collection activity being simulated within VIC. This "collector" may be a summarization or abstraction of real collectors modeled in VIC, with the summarization or abstraction performed during VIC output postprocessing. These observations are also removed from the MESSAGES queue by processing nodes within the communication network being modeled, while zero or more new observations (with differing EI, sender, sender-type, recipient, and time fields) are created and added to MESSAGES as a means of modeling the passage of information through nodes of the communication network being modeled.

If the additional message type "change-order" becomes modeled in future versions of PRO, they will be placed there by some process within the model to cause some change to a table or data structure representing the model, usually at some future (modeled) date/time. In this way, priorities, processing or communication delays, and other attributes of fusion nodes, communication links, or other entities within the model can be altered during its operation (for example, to reflect processing loads, focusing, time of day, what has been observed to date, how the war is going, etc.). We expect change-order types of messages to have the following additional fields:

Mesg-what-to-change An indication of the array or list entry to be changed

Mesg-new-value The replacement value for that entry

Unit Priorities

Some combinations of unit-ID and attribute (for example, the unit-effectiveness of unit E13000050) may be flagged by some commander as having high priority. Messages carrying observations about that particular unit-attribute combination should be given priority handling. To allow modeling of priorities, each unit-attribute pair is assigned an initial default priority (low or high). These priorities may be modified by the operation of rules during the running of the model (by use of eventual change-order types of messages placed into the MESSAGES queue). Conceptually, one can think of the priorities as existing in a unit-by-attribute 2D array:

Unit_priority[unit-ID, *attribute*]

each location of which contains a representation of either "high" or "low" indicating the priority of that unit-attribute combination.

In the current PRO model, certain unit priorities are established at the beginning of the run, and revised unit priorities are set (by use of a directive-type message) at time 01:19:00 (one day and 19 hours of simulation time into the run). Current unit priorities being used for these two time intervals are strictly notional and are representative of the type of qualitative information a corps commander would demand from his intel system.

High priority is given to the following categories of units, in each time period; all other units are given low priority.

High-priority units for period 1: from 00:00:00 to 01:18:00

—First echelon artillery assets
—All MAJOR command posts

High-priority units for period 2: from 01:19:00 to (end)

—3rd East German Army
—All MAJOR command posts

Decay of Unit-attribute Information Effectiveness

We assume various information regarding certain attributes of types of units decays at different rates. To record this, we store a table of decay factors (in real numbers representing decibels/hour) for each combination of unit-type (for example, cavalry, aviation-HQ) and unit-attribute. The information is represented by an information enhancement decay rate array:

Enhancement_decay_rate[unit-type, attribute]

The values stored in this table indicate initial default values. This information might be updated dynamically during the operation of PRO, but that feature is not implemented in the initial version of the model. At present, an enhancement decay rate of one decibel per hour is used for all unit-type/attribute combinations. These decay rate factors must be tailored to a particular application of PRO as explained in Sec. V.

During model operation, the EI value representing the enhancement of a commander's perception of a unit-attribute is decayed over a time interval DT by multiplying the appropriate entry in the enhancement-decay-rate table (divided by 3600 to convert it to per-second) by DT, then subtracting the result from the current value of EI (for that unit-attribute combination, for that node).

Order of Battle Databases

Any node in the communication net can maintain an order-of-battle database showing the values of its enhancement readings for each unit and attribute at various points in time. (Whether this database is maintained for a node is governed by the *Node-maintain-OOB* attribute of the "Node" structure.) This database is a collection of snapshots, each one being a table of unit versus attribute with an associated time_stamp. The entries in this table are enhancement values representing the enhancement or degradation of the commander's perception of the unit's attribute at this modeled time.

Each node designated as "maintain-OOB" is initialized to have an initial ORDER_OF_BATTLE database with "0" for each entry and a Time_stamp representing the starting time of the model. Subsequent snapshots are derived from this initial table by processing described below.

Each "maintain-OOB" node also maintains an associated MESSAGES queue of all messages received. These messages have the same form as described for observation-type messages above.

This MESSAGES queue for each recording node need not be a physical queue of such structures, it could merely be a list of pointers into a master queue of messages. The need for this individual node's MESSAGES queue can be eliminated if the order-of-battle database for this node is updated (that is, a snapshot is taken) every time this node receives a message. We believe that overhead would be too great, but this tradeoff analysis might be considered.

Weightings

We intend to use the PRO model to comparatively evaluate various combinations of intelligence collectors and processing (data fusion) regimes. To accomplish this, in the future we may want to develop a figure of merit for a particular model run (for a particular node's order-of-battle database within that run).

Because some unit-attribute combinations might be considered more important than others in developing this figure of merit, we need a table of weightings for unit-attributes, telling the relative importance of each of them in contributing to the final figure of merit. Therefore, we expect there will be many tables of this kind and tables that could easily be created on demand and could be applied to raw output from the model well after we ran it. These tables really are part of a postprocessor for PRO, to be run separately from the model, but are described here for completeness.

To this end, we envision a two-dimensional array of weightings:

Weightings[unit-ID, attribute]

where each entry in this table might have a number from 0 to 10, or from 0.0 to 1.0, to indicate relative weight to be given to a unit-attribute combination in the final figure of merit. There may be a requirement for a "family" of such weightings tables, reflecting different commanders' priorities, or something else. If it is decided that the weighting of a unit-attribute combination depends on the unit-type (for example, cavalry, aviation-HQ) rather than a specific unit's ID, then the weightings table could be much smaller.

The data contained in the ground truth database might contribute to the formation of these weightings, for example by having the weightings be a function of the battlefield location of a unit. The exact form of this relationship is to be determined. Weightings are not implemented in the present version of PRO.

MODEL CONTROL FLOW

Given the above data structures and data files, we can now describe the control flow of the PRO model as it routes messages among intelligence fusion processing nodes in the network being modeled.

At the most general level, the PRO model is exceedingly simple to explain: A list of "messages" initially consists of all the pre-observations generated by a VIC run. These messages are stored in date/time order, with the earliest first. The main PRO processing consists of the following general steps:

START:

1. Initialize data tables within the PRO model, place certain repeating directives into the message list to trigger periodic processes to be initiated, initialize the map display used as a "window" onto the model, and perform other initialization steps as needed.

The data tables being initialized by this step are those mentioned above, storing such information as the communication delay for each communication line for low- and high-priority messages, processing times (both low- and high-priority) for nodes, max_increment and elasticity values to be used in Eq. (6.6) for sensor/unit-attribute combinations, and so forth.

Repeating directives are special messages that, when processed, cause a copy of the message to be replaced on the message queue with a later date/time stamp. In this manner, periodic actions (such as update of the display) can be handled using the normal message processing mechanism of PRO.

LOOP:

2. Obtain the earliest message from the message list; if there are no more messages, perform cleanup activities (such as a final update of the display screen) and stop.

This most basic loop of the PRO model is governed by a procedure called "Process-mesgs." After reading the earliest message on the queue, this procedure determines whether the message is an observation- or a directive-type, and calls the appropriate subroutine for further processing.

3. If the message is a pre-observation from VIC, transform it into a PRO observation containing an enhancement increment.

Recall that pre-observations retain a FP # representing probability of detection for categorical attributes. If the current message is still a pre-observation from VIC, its associated probability_of_detection factor is transformed into an enhancement increment by use of Eq. (6.6), taking the values of max_increment and elasticity from a table of such values depending on the node (representing type of sensor) and unit-attribute observed. Equation (6.6) is also used for continuous attributes; in this case, the "probability_of_detection" term in Eq. (6.6) represents a standard error factor, and the stored values of max_increment and elasticity reflect that difference. For the remainder of the processing of this message and its successors (that is, messages derived from it as it passes through processing nodes), the enhancement increment term stored in the message, resulting from the above preprocessing, is used directly in succeeding processing steps.

4. If the message is an observation (or has been transformed into one by the previous step) then begin to act on the message on behalf of the recipient node (N1). This node now becomes the new "sender." Then:

   4.1. If node N1 maintains an order-of-battle database, then update node N1's enhancement value for the unit-attribute combination mentioned in this message by adding to it the EI contained within the message. The next time the "decay enhancements" directive is processed, this enhancement value is decayed by the product of the appropriate decay factor and the time interval since it was last decayed.

Any node can be flagged as maintaining an order-of-battle database, meaning that a table of enhancement values, one for each unit-attribute combination, is stored for it. As an observation message is received at the node, the relevant unit-attribute enhancement is updated using Eq. (6.5b). That is, the enhancement increment in the message is merely added to the existing enhancement value. (All enhancement values are initialized to zero at the beginning of a model run.) In addition, Eq. (6.7) is periodically used to decay the enhancement value. These processes are mainly carried out by the procedure "Process-observations."

4.2. Add to the message's date/time stamp to account for processing delay in node N1.

The array "Unit_priority[unit-ID, attribute]" contains "high" or "low" designations for each unit-attribute combination. For the unit-attribute mentioned in the current message, the priority is obtained; based on this priority, the value of a processing delay for the current node is contained in the node's attribute "Node-default-process-delay."

4.3. For each node, N2, in the communication network to which node N1 sends messages of this type, perform the following steps. (Thus a given message can cause zero or more messages to be propagated.)

4.3.1. Create a copy of the original message; in the copy, make the current node (N1) the sender and N2 the recipient node

4.3.2. Add to the copied message's date/time stamp to account for communication delay from node N1 to node N2

4.3.3. Place the message copy back into the master message list in sequence based on its (new) date/time stamp, so that the message list retains its ordering by message date/time

The communication delay from node N1 to N2 is found in array "Default_comm_delay," and the restrictions in array "Propagate_att" are checked to see if this is a valid message. If not, the new message is not sent. The set of all nodes to whom N1 communicates is all those in the "Default_comm_delay" array for which N1 is the from_node, and the delay time to any to_node is not "infinite."[15] These actions are coordinated by the procedure "Process-observation," calling upon other subroutines as needed.

4.4. Delete the original message from the message list and go to LOOP.

The incoming observation has been "processed" by the current node (N1) and has spawned zero or more other messages to other nodes. The incoming message has no further purpose and is deleted from the message list.

5. If the message is a directive (such as update display, or write the order-of-battle databases to disk), then execute the directive. If the directive is a repeating one, increment its date/time stamp and replace the directive in the message list, so that it will "fire" at a future date/time. Go back to LOOP. (Note: one repeating directive that is placed into the message list during initialization of the model is one that triggers a decay of all enhancement values for all unit-attribute combinations at each node for which order-of-battle databases are being maintained, to reflect the decay of the accuracy of information with the passage of time. The rate of decay used depends on the particular type of unit and unit-attribute.)

The directive-type messages currently processed by PRO are: read-truth; stop; log-OOB; decay-enhancements; update-all-unit-icons. The actions associated with these directives are mostly self-explanatory and mentioned in the subsection "Messages," above.

USER INTERFACE/DISPLAY MODULE

The purposes of the user interface module are to (1) display the actual (ground truth) locations of some or all of the Red and Blue units on a map, as a function of simulated time, so that the user can view the military situation; (2) display the color intensity of each icon representing a Red unit to represent the current enhancement value *for some attribute* of that unit, as recorded by *one node* in the communication network for which an order-of-battle file has been generated; and

[15] We put "infinite" in quotes because it is represented within the computer by a very large but finite number.

(3) allow the user to query various data in the system and provide other interactive control over the operation of the PRO model.

Figure 13 illustrates a typical map display for a section of the U.S. V Corps area of interest in Central Europe. In this figure, the color intensities recording enhancement values for some unit-attribute have been displayed as grey levels.

To perform its functions, the user interface module accesses the order-of-battle database (being) generated for a particular node in the communication network. This OOB database contains "snapshots" of data at regular intervals (for example, four-hour), with these snapshots including both ground truth data and all unit-attribute enhancement values for that simulated date/time.

The user interface module has been constructed from a program called "Map Tool" written for the RSAS. Map Tool is a display program that knows about maps and geometric forms, but not about geography, specific maps, tabular displays, or military substance. The software interface between Map Tool and PRO is called the Application Interface Processor (AIP); the AIP knows which maps are wanted and how to form tabular displays about a place or item on the map based on data from the rest of PRO. The design of the AIP has been strongly influenced by, and borrows portions of its code from, the RSAS Data Editor system.

Among the control actions available to the user are:

- Click the mouse within the "scroll bars" at the bottom and right edge of the map display to scroll the map left/right and up/down
- Select a unit icon (by clicking the left mouse button while the cursor is nearest to that icon, within some limit) and display various unit-attribute enhancement values or ground truth data items for that icon, as stored by one or more network nodes
- Move a tabular display of data for a unit to a convenient place on the display screen.

Figure 14 shows a pop-up tabular display of unit-attribute enhancement values on the map display after a particular unit is mouse-selected.

Many other interactive features are available to the user, such as on-line help facilities, tailoring of the tabular displays shown upon selection of an icon, accessing other tabular data displays showing other data values within the PRO system, map area coloring. These features are described in a help file available to the PRO model user.

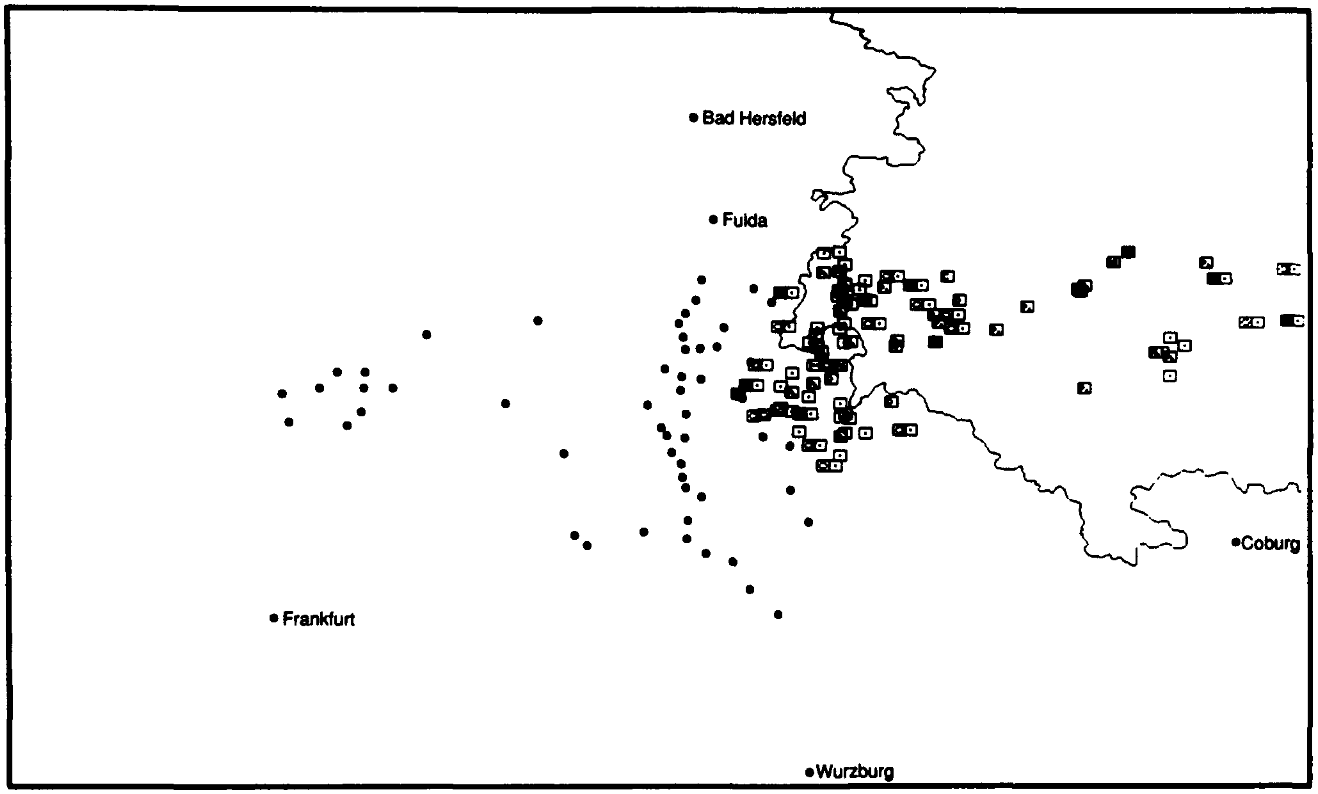
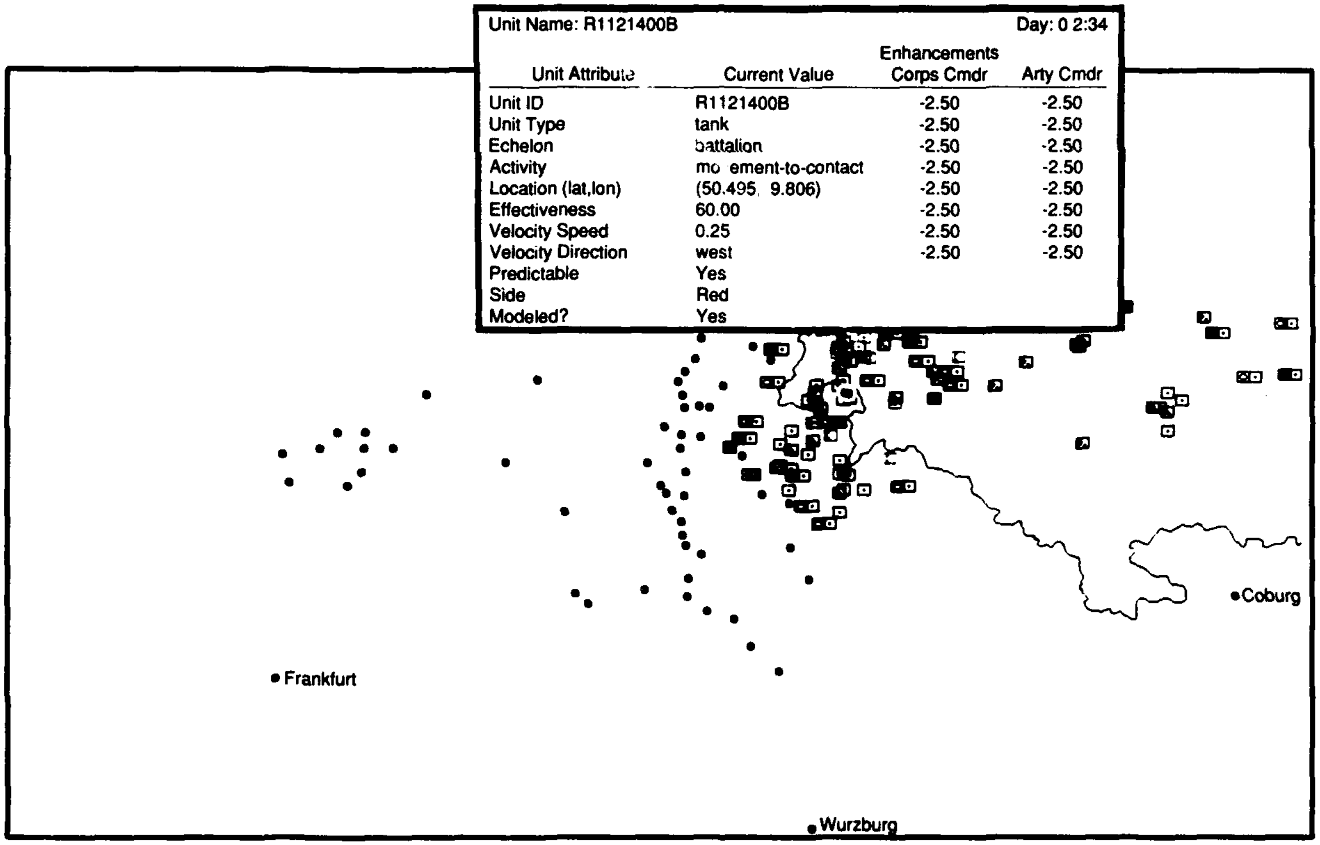


Fig. 13—User interface map display for PRO model



Unit Name: R1121400B Day: 0 2:34

		Enhancements	
Unit Attribute	Current Value	Corps Cmdr	Arty Cmdr
Unit ID	R1121400B	-2.50	-2.50
Unit Type	tank	-2.50	-2.50
Echelon	battalion	-2.50	-2.50
Activity	mo ement-to-contact	-2.50	-2.50
Location (lat,lon)	(50.495, 9.806)	-2.50	-2.50
Effectiveness	60.00	-2.50	-2.50
Velocity Speed	0.25	-2.50	-2.50
Velocity Direction	west	-2.50	-2.50
Predictable	Yes		
Side	Red		
Modeled?	Yes		

Fig. 14—User interface map display with tabular data overlay

In the present model, each unit icon on the map display is shown in one of five color intensities (from light pink to dark red), to graphically represent the enhancement value for some attribute of that unit (usually location, but not necessarily) at that simulated date/time. The four enhancement threshold values used as boundaries between these five categories of values are determined separately for each attribute. When PRO is used for intelligence fusion analyses, these boundaries can be chosen to highlight particular ranges of data quality readings for a unit-attribute of interest to the analyst. At present, these threshold values are chosen so that, for all the units being displayed and for this particular attribute and when integrated over time for the entire PRO model run, an (approximately) equal number of units will be placed in each of the five color intensity categories.

It is possible to run the user interface to PRO either in real time as PRO operates, or else in a retrospective "movie mode" based on the files generated during a PRO model run. Note that even in the movie mode, the user has substantial control over the user interface, for example in selecting unit-attributes to be viewed, mapping of data quality values into color intensities, and selection of units and map portion to be viewed.

PRO OPERATING ENVIRONMENT REQUIREMENTS

At present, the PRO model runs on a Sun 4 workstation (from Sun Microsystems) with color monitor, under the Sun-UNIX[16] 4.01 operating system and the SunTools interface. A minimum of 16 megabytes of RAM memory is required, and we recommend at least 100 megabytes of disk space to hold the VIC output files being processed, order-of-battle files generated during a PRO run, and executable code for PRO, Map Tool, and the auxiliary programs they require. The above size requirements do not include space for the storage of the operating system, SunTools, and other system packages.

On a Sun 4/110 (8 MIPS), PRO currently operates at approximately 25 seconds of wall clock time per simulated hour. The code has been somewhat optimized, but we expect that considerable improvements in operating efficiency can be made as time and resources permit.

[16] UNIX is a trademark of Bell Laboratories.

SUMMARY

We have developed a computer model called PRO to study the effects of the change in quality of intelligence information at various nodes within an intelligence fusion network given sensor observations of various unit-attributes and the decay in the quality of intelligence information as time passes. We are currently using the VIC simulation to generate movements of units on a battlefield and sensor observations of various attributes of those units.

The PRO model has considerable flexibility. For example: (1) it is not dependent on VIC as the battlefield simulation; another program could be substituted without much difficulty; (2) the battlefield units being modeled can be selected; (3) the sensor readings to be modeled are governed by tabular entries that can be changed; (4) the configuration of the communication network linking intelligence processing nodes can be changed, as can attributes of the links and nodes (such as their transmission or processing delay times); (5) the change in the quality of intelligence information can be assessed at any node in the communication network, merely by flagging that node as one that maintains an order-of-battle database; and (6) during model execution, much of the information within the model can be inspected at an interactive map display user interface, and many system parameters can be changed during the model run.

As its main processing cycle, PRO traces the passage of "messages" (mostly representing sensor observations) through the network of processing nodes, computing the increase or decrease in quality of information about unit-attributes at each node. In general, sensor observations increase quality of information (although this need not be so) and the quality decays with the passage of time. The quality of information about unit-attributes at a node is measured only relative to the starting point (= 0), not in absolute terms.

Most of the PRO model is written in the RAND-ABEL language, which was designed to permit readability of models by persons with knowledge of the subject matter being modeled. PRO has also utilized major features of the RSAS, particularly the Data Editor and a map graphic display package (Map Tool). PRO currently operates on a Sun 4 workstation under Sun-UNIX version 4.01.

VII. CONCLUSIONS

This report describes a new way to evaluate intelligence systems. It has a distinctive set of features that differentiate it from alternative approaches. This section summarizes these features, then reviews factors that analysts should consider as they validate our approach in a particular application and suggests applications in which the approach should prove useful.

KEY FEATURES OF THE APPROACH

Because the evaluation approach offered here is forward-looking, it relies heavily on simulation and on the comparison of simulated alternatives. Our approach to evaluation and to simulation differs considerably from other approaches because we have carefully crafted our approach to help analysts examine a specific question: *How do specific, incremental changes in a combat intelligence system affect the quality of relevant information available to decisionmakers on the deep battlefield?* Our pursuit of a technique that analysts could use to answer this question led to choices that make our approach different from others in six important ways.

1. *We evaluate intelligence development by using a figure of merit based on an explicit definition of the quality of intelligence products. We focus on one intelligence product in particular, the Red order of battle in the deep battlefield.* Other approaches look at the technical performance of particular parts of an intelligence system, time lines for delivering information from the battlefield to a decisionmaker, and the effect of intelligence development on combat outcomes. All of these measures are valid and useful in particular applications. Our measure is best for looking at the performance of the intelligence system as a whole and in depth without having to determine how it interacts with other sources of combat capability.

2. *We focus on incremental changes in intelligence systems.* Our approach allows us to examine how certain changes in collectors, processors, and communication links affect the total performance of an intelligence system. Focusing on incremental changes allows us to avoid the ambiguities involved in modeling important feedbacks within an intelligence system and between the intelligence system and other combat capabilities. For example, we need not posit assumptions about how information flows affect delays in communication and processing,

how information available today affects the demand for information tomorrow, or how changes in the quality of information affect combat outcomes and hence the nature of Red behavior in the future. This last point also means that we need not posit assumptions about how an intelligence system transforms data into higher-level inferences and how these inferences affect command decisions. Because we need not speculate about these factors, where great uncertainties reside, we can examine issues where more is understood and use these to draw more easily defensible conclusions about the performance of combat intelligence systems. Where feedbacks like those that we avoid are important to policy, however, a user should consider an alternative approach that looks beyond the effects of incremental changes.

3. *We rely heavily on Army models for input.* As currently formulated, our approach relies heavily on the Army's VIC corps combat model to simulate the behavior of Red units on the deep battlefield and aspects of a Blue intelligence system's collection of data on this behavior. With modest modifications, we could accept such information from alternative sources; to our knowledge, no other sources can provide the depth of detail that VIC provides and that we need to provide the richness we seek in our own simulation. Given its status as the Army's approved corps combat model, VIC embodies Army doctrine in a way that no other available model does. We can and have adjusted inputs from VIC in small ways that do not challenge Army doctrine. Users who prefer to use a combat simulation other than VIC can potentially use their own combat simulation to drive our simulation if it generates suitable information. They will need to provide considerable substantive information on the behavior of Red units and on Blue's collection of information about these units that we do not address directly as part of our approach.

4. *We simulate the quality of intelligence products, not the generation of these products per se.* An intuitively appealing way to present information about the performance of an intelligence system might be to simulate the Red order of battle as Blue intelligence perceives it, compare this perception with the true Red order of battle, and use the difference between the two as a figure of merit. We rejected this potentially attractive approach because simulating a perceived Red order of battle would require massive detail about specific fusion rules and strong assumptions about higher-level inferences about Red behavior; such a simulation cannot disentangle the order of battle from these higher-level inferences. Past attempts to simulate a perceived Red order of battle have yielded enough questionable inferences to undermine confidence in these simulations. We have been able to develop a method that simulates the quality of intelligence directly

without attempting to simulate specific perceptions or make assumptions about the inferences that make it so hard to simulate these perceptions. Our approach makes strong assumptions about the loss functions of decisionmakers who use data on the Red order of battle, but we believe the simplicity that these assumptions justify our decision to use them.

5. *Our simulation of intelligence fusion takes a high-level approach to avoid getting lost in the intricate detail of true fusion.* As a result, our approach does not attempt to collect rules that order-of-battle analysts and automated systems use to execute fusion and provide an inference engine that executes these rules together. Attempts to do this in other settings have not yet succeeded. We take a more aggregate approach based on a set of intelligence concepts and parameters that we have not seen in earlier simulations of intelligence development. For example, while past efforts have typically used a probability of detection to measure the quality of information yielded by collection, our approach uses a discrimination ratio or enhancement increment, concepts we find useful because of their analytic power and their ability to capture basic ideas that underlie the detailed rules of thumb used in true fusion. Because other analysts have not used these concepts in the past, no one has attempted to collect data to measure them. Similar statements could be made about other concepts that we use. This complicates the immediate implementation of our approach. If, as we expect, our approach simplifies the effective simulation of fusion at a high level, we expect the data we need to become more accessible, making our approach easier to implement as time passes.

6. *We implement our approach with code that promotes easy understanding and modification to include new rules as needed.* We have written substantive portions of the code in the C-based English-like language RAND-ABEL, which allows users with little programming experience to look directly at the implemented code and understand what it is doing. The structure of the code makes it easy to change collectors, processors, communications links, and the way that they interact in an intelligence system. Its clear, modular form also allows targeted adjustments in the code if specific new rules are required to characterize specific capabilities that are important to a policy evaluation in an intelligence system. Using RAND-ABEL also gives us access to the editing and graphics utilities of the RSAS, which greatly facilitates the use of our approach and interpretation of the output that it generates. Where computational speed is more important than easy comprehension of the substance of our model, we have used C to increase this speed. Other simulations that give less importance to comprehension run faster than the code we have devised. We believe

that such comprehension is vital in this model because we expect analysts who test sensitivities and examine policy alternatives to change the model often. Comprehension is important enough to justify the slower run times required to support it.

VALIDATING SPECIFIC APPLICATIONS OF THE APPROACH

The next step in using this approach will be to select a specific policy problem and validate a simulation of the intelligence system relevant to this problem. Because intelligence development is such a complex process, we have not attempted to capture the full structure of each activity that contributes to intelligence development. On the contrary, we have attempted to specify a minimal set of parameter values that we can use to capture the performance of the system as a whole and the effect on total performance of changing selected aspects of the intelligence system. As a result, we cannot validate our approach simply by choosing realistic values of inputs and accepting whatever output the approach generates. Instead, users must be prepared to develop a baseline case that generates reasonable *outputs* and use that as a starting point to ask how changes in the baseline would affect the quality of information in the intelligence system. Developing such a baseline will take care and patience.

In this regard, our model is not different from other simulations of intelligence development that the Army currently uses for training purposes. When one asks users of these systems how they validate these simulations, they invariably respond that they adjust the simulations until they generate reasonable results. Our approach must be adjusted in a similar way.

What is the meaning of "reasonable results?" The simulated quality of a commander's information should vary in certain systematic ways in the baseline case. It should generally fall as Blue looks farther beyond the FLOT. It should abruptly increase following a collection mission that brings new information into the intelligence system. Quality should vary systematically across attributes. Blue should generally know more about unit name, type, and echelon than about unit effectiveness or activity. Quality should vary by unit type. Blue should know much more about the location of major units with armored vehicles than about the location of surface-to-surface missile batteries. The degradation factors that we calculate should be higher for some attributes than for others—for example, higher for location than for unit identification. And in general, the quality of information should be high at the beginning of a war, fall over a period of time after the war starts, and then, if the scenario lasts long enough, recover as collection and processing activities intensify and Red behavior becomes more predictable. These are macrotrends we would want to observe in the baseline; an experienced order-of-battle analyst could specify a more detailed set.

Once a reasonable baseline case is developed, sensitivities should also be used to ensure that the model reacts to parameter changes in reasonable ways. For example, removing JSTARS should degrade the quality of information on unit location without affecting information about unit identification much. Changing the delay time on the quick-fire channel that links an MTI collector with an artillery commander should not affect the contribution of COMINT internals to the quality of a corps commander's information on the identity of units.

These sensitivities can look very much like the changes in an intelligence system whose effects we wish to examine. In fact, the typical process of using a model of this kind is one of simultaneously validating the model and developing the results of policy analysis. We should not expect the model simply to generate useful numerical results without a careful examination of how the model generated these results and how they might change if the model were specified differently. The model's purpose is essentially to assist an analyst in ensuring a reasonable story to explain why a change in an intelligence system has the effects it has. Again, an experienced order-of-battle analyst should play an integral part in this process. The model is specifically designed to allow a user to examine the order of battle maintained at any node during an engagement; this capability should allow an analyst to develop a fairly subtle understanding of information flows in the model.

As the discussion of the simple intelligence system that we use to illustrate points in Secs. III through V should indicate, we have chosen the values of parameters for that system to achieve results like those suggested above. But the proof is in the pudding. In all likelihood, it will take a good deal of adjustment with a realistic, complex intelligence system to choose a set of input parameters that achieves a reasonable baseline and hence prepares the model for useful analyses of changes in an intelligence system.

It may well be easy to abuse this approach by manipulating its parameter values until they yield results that support a predetermined policy position. That is a risk associated with all combat simulations of this kind. We attempt to limit such abuse by requiring careful attention to the behavior of the simulation in the baseline case. That is, the simulation must produce reasonable results before any policy change is considered. Nonetheless, an unscrupulous user with sufficient resources will probably be able to prepare a baseline case in a way that yields the "right" results when policy excursions are modeled. In the end, this approach will yield the most useful new information for analysts truly interested in getting it. Such analysts will not dismiss unexpected results out of hand. They will instead seek a commonsense explanation. When they can satisfy themselves that such unexpected results make sense, they will benefit from the approach's ability to yield new insights.

FUTURE APPLICATIONS

The discussion in the text focuses on evaluating how specific changes in an intelligence system would affect the quality of combat intelligence on the deep battlefield in a Central European war. This is the issue that we had in mind when we designed the approach. It is not the only way that the approach might be used.

The most obvious opportunities are evaluations of intelligence systems that cover more than the deep battlefield and that operate outside Europe. Moving beyond the deep battlefield in Europe simply involves including additional collection, processing, and communication assets to reflect expansion into the close or, potentially, the rear battle area. VIC supports all collectors relevant to a corps area of interest, including division assets relevant to the close battle. Including these assets would slow the execution of simulation runs by increasing the number of computations required to complete a simulation. We will not know how serious a problem this presents until actual applications are made. We have explored the use of sampling—tracking information about only selected Red units on the battlefield—as a way to reduce computation times, and that looks promising; the results do not depend on the number of Red units actually included.

Moving beyond Europe presents a more serious challenge. VIC scenarios exist only for the European theater and then only for selected U.S. corps areas in Germany. To move beyond these areas, a user would have to find or develop a suitable substitute for VIC and adapt our model to that substitute. Once that was done, the approach should operate with little difficulty.

Once we move beyond dependence on VIC, other opportunities open up. For example, the approach could be used to simulate peacetime intelligence development. If intelligence assets were exercised as part of a large-scale Blue field exercise, analysts could measure the performance of the intelligence system against unit behavior that was well documented. Analysts could use our approach to simulate the performance of this intelligence system and, by comparing simulated with actual performance, learn more about how to select values for important parameters in our simulation. Our approach might also be used to help evaluate the U.S. peacetime program for monitoring military activities in eastern Europe. With a suitable driver, it could be used to examine strategic intelligence development. In such an application, a different set of unit-attributes would presumably become important; our approach could easily be adapted to deal with this. We have not explored these possibilities in any depth, so we cannot comment on the availability of suitable drivers or the difficulty of adapting our approach to these drivers. Our main point is to emphasize the flexibility that lies at the core of our approach.

The ability to evaluate incremental changes in intelligence systems raises the possibility of a very different kind of application. Let us examine it in the context of the European-theater, corps-level intelligence system that we emphasize in the text. The depiction of intelligence in current combat simulations is fairly crude. Our approach could potentially provide a basis for enhancing the treatment of intelligence. The approach runs too slowly to be incorporated as an integral part of these simulations, but analysts could use it to generate tables or parameter values integral to combat simulations.

For example, a combat simulation might characterize a corps intelligence system in terms of the presence of certain key assets or a general statement about the level of quality of key factors. These factors might be identified in terms of collection, processing, and communications; intelligence disciplines such as COMINT, ELINT, and IMINT; intelligence functions such as situation assessment and target acquisition; or some other scheme. Analysts could then structure a real intelligence system that yielded the levels of quality associated with these factors and use our approach to simulate it. By varying appropriate elements of the system, these analysts could simulate the way changes in their quality factors changed the quality of intelligence relevant to the combat simulation. Such an application steps well beyond our approach and requires careful consideration of many factors not discussed. But the approach we offer provides a way to develop such information for combat simulations.

Other applications might also be considered. The next logical step, however, is to do something "simple." Analysts must apply the approach to a real intelligence system, validate the application to ensure that it yields reasonable results, and use it to evaluate selected changes in that system. The process of applying the approach in this way will tell us a great deal more about the capabilities that this approach offers than we can specify now with any confidence.

Appendix

DECEPTION, GHOSTS, AND PREDICTABILITY

Soviet doctrine gives high priority to the use of deception as an integral part of military operations. One of the principal goals of Blue intelligence must be to detect deception and to sort out what Red is really doing from what he wants Blue to believe he is doing. Our approach is designed to analyze two aspects of Red deception and of Blue intelligence's ability to deal with it. VIC does not generate the information we would need to implement these aspects of our approach. But we have designed our model so that they could be implemented as soon as suitable information was available.

The two aspects of deception that we examine are the presence of "ghost" units and the predictability of the behavior of Red units on the deep battlefield.

GHOSTS

We calculate the quality of information that Blue intelligence maintains on specific attributes of each Red unit on the deep battlefield. However, when Blue intelligence makes an error, it is not always useful to think of it in terms of a reduction in the quality of Blue information on units that Red actually employs. For example, when Blue posits Red units that do not exist, it may be important to understand how strongly Blue holds its beliefs about these units and how changes in an intelligence system affect these beliefs. We refer to such Red units as "ghosts."

Several factors can lead Blue to believe in ghosts. The notion is fundamental to the simplest forms of Soviet *maskirovka*, which use radio traffic and simple replicas of tanks and other heavy vehicles, equipped with devices that produce realistic visual and thermal "signatures" for these vehicles, to create the illusion that a unit is at a location when it is not. But Red need not attempt to create ghosts for Blue to perceive their presence. For example, emanations from two separate but similar radar installations can lead Blue to believe a radar installation lies somewhere between them. That can happen whether Red intends it to happen or not. In each of these cases, Blue infers the presence of a Red unit that does not exist. How can we analyze the effect of changes in an intelligence system on such inferences?

We reserve a unit-attribute that describes whether a Red unit on the battlefield is real or not and measure the subjective probability weight that a Blue intelligence system places on the right value of this attribute. In VIC, all units currently portrayed are real. But, using VIC's data input preprocessor, we could add ghost units to the scenario that VIC portrays, together with details about their behavior on the deep battlefield. We could then specify simple parametric models like those in Sec. IV to show how values of VIC's "probability of detection" translate into discrimination ratios relevant to this unit-attribute. This exercise would require the special knowledge of an order-of-battle specialist familiar with Soviet doctrine and the circumstances under which Blue collectors perceive ghosts on the battlefield. We have designed our model to accept such information as soon as it is developed.

PREDICTABILITY

In Sec. IV, we suggest that it is desirable to reflect the following rule in our simulation of intelligence development:

> The quality of Blue intelligence on a particular Red unit-attribute rises as Blue's ability to predict the behavior of that unit-attribute rises.

We capture this rule to some extent in the degradation factors described in Sec. IV and in factors that convert VIC data into discrimination ratios in a way that reflects Blue's relative ability to maintain accurate models of different kinds of unit-attributes. But in certain circumstances, Red can deliberately deploy a unit in a way that runs counter to Blue expectations. If Blue is not looking for the kind of behavior that a Red unit is pursuing, Blue is less likely to observe this behavior accurately. How can we capture the effects of changes in an intelligence system on the quality of Blue's information about Red units that do not behave as expected?

We do this by reserving a unit-attribute for each Red unit on the battlefield that states whether that unit is behaving "predictably" or not. The value of this unit-attribute acts as an input to a simple rule that adjusts the data we receive from VIC. If the unit is behaving predictably, this rule does not change the input from VIC. If the unit is not behaving predictably, the rule degrades the information from VIC. To implement this feature, we rely on an experienced order-of-battle analyst to add information to our input file identifying each sighting in which a unit-attribute is not behaving predictably. We must then determine how much to degrade information. That is, we expect the determination of what behavior is predictable and how predictability affects the input to PRO to be made off-line, without formal rules, by an informed order-of-battle analyst. Although we have not implemented this feature, the model is currently written to implement it as soon as appropriate information is available.

These are not the only forms of deception that might occur on the deep battlefield. In fact, Red's real goal is to achieve operational deception with regard to higher-level inferences about its basic intent. As explained in Sec. II, we do not attempt to analyze an intelligence system's ability to develop accurate higher-level inferences. The forms of deception discussed here are relevant to the Red order of battle that we emphasize. They should be understood in that context.

BIBLIOGRAPHY

Blackman, Samuel S., *Multiple-Target Tracking with Radar Applications*, Artech House, Norwood, Massachusetts, 1986.

Bunn, Derek W., *Applied Decision Analysis*, McGraw-Hill, New York, 1984.

Davis, Paul K., Stephen C. Bankes, and James P. Kahan, *A New Methodology for Modeling National Command Level Decisionmaking in War Games and Simulations*, The RAND Corporation, R-3290-NA, 1986.

Gamble, Allen E., et al., "Vector-in-Commander (VIC) Documentation: Data Input and Methodology Manual," TRAC-WSMR-TD-Draft, U.S. Army TRADOC Analysis Command, White Sands Missile Range, New Mexico, 1987.

Hogg, Robert V., and Allen T. Craig, *Introduction to Mathematical Statistics*, Macmillan, London, 1970.

Kahan, James P., "Air Support in CENTAG's Deep Operations," *Military Review*, Vol. 64, August 1989, pp. 64–73.

Kahan, James P., D. Robert Worley, and Cathleen Stasz, *Understanding Commanders' Information Needs*, The RAND Corporation, R-3757-A, 1989.

Raiffa, Howard, *Decision Analysis: Introductory Lectures on Choice Under Uncertainty*, Addison-Wesley, Menlo Park, California, 1968.

Ramsey, Connie L., et al., "A Comparative Analysis of Methods for Expert Systems," *Int. J. Man-Machine Studies*, Vol. 24, 1986, pp. 475–499.

Shapiro, Norman Z., et al., *The RAND-ABEL Programming Language: History, Rationale, and Design*, The RAND Corporation, R-3274-NA, 1985.

Shapiro, Norman Z., et al., *The RAND-ABEL Programming Language: Reference Manual*, The RAND Corporation, N-2367-1-NA, 1988.

U.S. Army, Headquarters, Department of the Army, *Intelligence and Electronic Warfare Operations*, Field Manual 34-1, Washington, D.C., 1987.

Wohlstetter, Roberta, *Pearl Harbor: Warning and Decision*, Stanford University Press, Palo Alto, California, 1962.

Zellner, Arnold, *An Introduction to Bayesian Inference in Econometrics*, Krieger, Malabar, Florida, 1987.